2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Virtual model control for compliant reaching under uncertainties

Yi Zhang[1], Daniel Larby[1], Fumiya Iida[1], and Fulvio Forni[1]

Abstract— **Virtual Model Control (VMC) is an approach to design a controller for force-controlled robots in complex uncertain environments. While this method was primarily investigated for legged robot locomotion in the past, it can be more generally applicable to other types of robotic systems. This paper investigates the VMC framework for reaching tasks in a force-controlled robotic arm. We propose six different approaches to designing virtual models in order to achieve reaching tasks in environments with obstacles and uncertainties. A force-controlled 8 degree-of-freedom humanoid robot was used to validate the proposed approach in the real world. We conducted three experiments to test the performance of VMC controllers in terms of predictability, sensitivity to external force, and adaptability against known and unknown obstacles. Experimental analyses show that, even though the proposed approach needs to sacrifice accuracy and trajectory optimality, it enables us to design complex reaching motions under uncertainties, in an intuitive and extendable manner.**

I. Introduction

Safe planning and control of robotic arms for reaching tasks in uncertain cluttered environment are still unsolved problems. The fundamental issue lies in the inability of traditional motion control and planning methods to negotiate in real time with real-world scenarios, which are inherently unstructured and where object shapes and obstacle locations are only partially known. In fact, the remarkable speed and accuracy of our best industrial robots require a well-known and restricted factory floor environment.

The conventional solutions typically employ an interplay of complex sensing, state estimation, planning, and feedback control, but most of them require complex and computationally expensive sensing and planning processes, which lack interact-ability to accommodate uncertainties. Furthermore, the best planning methods such as Probabilistic Road Map (PRM) [1], Rapidly Exploring Random Tree (RRT) [2], Potential field (PF) [3] are then often realised through inverse kinematics and joint-space high-gain PID control. This makes the robot stiff to guarantee accurate tracking, at the cost of low sensitivity to external perturbations, such as unknown obstacles. As a result, conventional solutions show limitations in cluttered environments, characterized by inaccurate information.

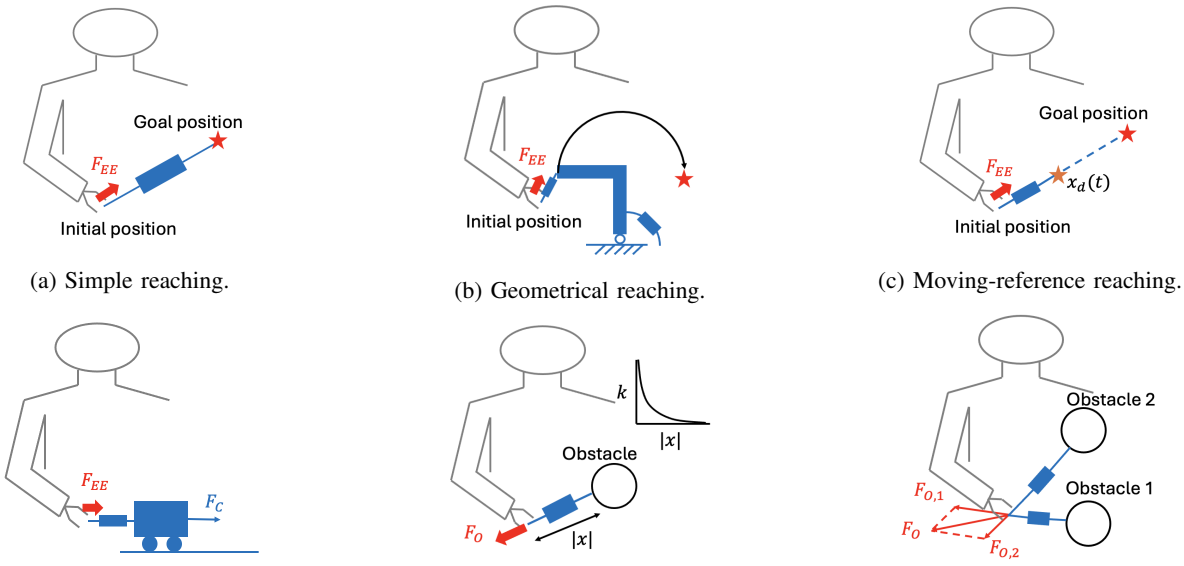
This paper advocates for methods that integrate motion planning and motion control to handle uncertainties in a safe way. Ideally, we want to take advantage of the available information on the environment and also offer a certain degree of compliant motion for real-time adaptation to unexpected scenarios. We propose the use of *virtual model control* [4] for the safe, intuitive, and tractable approach to plan and control reaching tasks for articulated robotic arms. In virtual model control, the robot behavior emerges from the interaction between the mechanics of the robot interconnected with designed 'virtual' mechanism, given by a collection of springs and dampers organized in a suitable geometry. Eventually, this 'embodied' controller is simulated in a computer and realized through current/torque commands to the actuators. The virtual mechanism pulls the robot behavior into specific patterns, leaving a certain degree of sensitivity to the interaction with the environment. The robot steers in the direction defined by the total energy of the system, shaped by virtual mechanisms and environment. Therefore, trajectory planning and motion control are integrated.

Our hypothesis is that virtual model control offers a practical solution to the challenge of safe motion planning and control, by negotiating between various task specifications in complex environments. Specifically, this paper proposes six distinctive ways to design virtual models for the purpose of reaching in uncertain cluttered environment, illustrated in Fig. 1. *Simple reaching* is the most intuitive motion planning approach. It shows similarities with PD control and impedance control in end effector space [5], [6]. *Geometric*, *moving-reference*, and *force-driven reaching* extend motion planning with additional features, leading to a higher degree of accuracy and design flexibility. *Obstacle* and *multi-obstacles avoidance* consider how to reach the target location while avoiding known obstacles.

We highlight two key features of virtual model control: (i) *predictability* of (controlled) behavior and (ii) its *intuitive design*. Virtual model control makes the action of the controller explicit. The resulting control algorithms are predictable because they correspond to a mechanical interaction driven by physics. By inspecting the virtual mechanisms and the environment where the robot operates, we have an expectation of the intended motion, which makes the controller easier to understand and debug. Furthermore, given the planning requirements, the design of the controller is as simple as placing virtual components and tuning their parameters. This allows one to tailor the design of the controller to the specific features of the task. Through suitable tuning of the (virtual) mechanical parameters, the robot remains compliant. The force exerted when a reaching cannot be achieved is controlled by the virtual stiffness of virtual mechanical springs. That is, virtual model control trades motion accuracy for controlled sensitivity to the interaction with the environment. The controlled robot also enjoys the benefits of passivity [7],

*Y. Zhang is supported by Sciences Research Council and AgriFoR-wArdS CDT EP/S023917/1. D. Larby is partially supported by the EPSRC grant EP/T517847/1 and by CMR Surgical.

[1]Department of Engineering, University of Cambridge, UK.



(a) Simple reaching. (b) Geometrical reaching. (c) Moving-reference reaching.

(d) Force-driven reaching. (e) Obstacle avoidance. (f) Multi-obstacles avoidance.

Fig. 1: (a)-(f) Virtual mechanisms designed for satisfying different planning goals.

i.e. stability and robustness, and controlled impedance [8].

The paper is organized as follows. The virtual model control approach is discussed in Section II through examples. The robot platform and implementation details are provided in Section III. The experiments are discussed in Section IV: (i) unperturbed reaching, (ii) reaching with known obstacles, and (iii) reaching with unknown obstacles. Further discussions and conclusions are provided in Section V.

II. Virtual Model Control

A. Virtual model, virtual forces, and motor torques

A virtual model controller is a collection of springs and dampers organized in a mechanical structure and connected to the robot. In this section, we show examples of virtual model controllers for reach motion and obstacle avoidance. Each controller is a virtual mechanism generating driving virtual forces, which are then translated and realized by robot actuators using torque control. The translation can be obtained using the robot geometry.

For fully actuated robots, using $\boldsymbol{q}$ to denote the joint coordinates of the robot, the force $\boldsymbol{f}_i$ generated by a virtual component at a point $\boldsymbol{h}_i(\boldsymbol{q})$ of the robot is mapped into corresponding joint torques $\boldsymbol{\tau}$ using the Jacobian matrix $J_i(\boldsymbol{q}) = \frac{\partial \boldsymbol{h}_i(\boldsymbol{q})}{\partial \boldsymbol{q}}$. Following [9], the torque command to generate all the forces of the virtual model controller is given by

$$\boldsymbol{\tau} = J_1(\boldsymbol{q})^T \boldsymbol{f}_1 + J_2(\boldsymbol{q})^T \boldsymbol{f}_2 + J_3(\boldsymbol{q})^T \boldsymbol{f}_3 + \dots \quad (1)$$

Virtual model control is inherently stable, since the controlled robot is just the interconnection of the passive mechanics of the robot with the passive virtual mechanics. Stability is guaranteed by passivity theory [10], [11], [12]. In fact, virtual model control provides an intuitive, task-oriented method to do energy shaping and damping injection in robotics [7], [13]. The selection of spring and damper parameters and their geometric configuration also shape the impedance of the robot [8], [14].

B. Simple, geometrical, and force-driven reaching

In its simplest form, moving the end effector to a desired position $\boldsymbol{x}_d$ is achieved by a spring with stiffness k and a damper with damping c attached from the end effector position $\boldsymbol{x}_{EE}$ to the desired position. These components produce the virtual force

$$\boldsymbol{f}_{EE} = k\boldsymbol{x} + c\dot{\boldsymbol{x}},\ \boldsymbol{x} = \boldsymbol{x}_d - \boldsymbol{x}_{EE}. \quad (2)$$

when the end effector deviates from the desired position and velocity, as shown in Fig. 1a. This corresponds to the classical proportional derivative control in the end-effector space. Orientation can be controlled in a similar way, through rotational virtual spring and dampers.

In more complicated reaching scenarios, such as those involving geometric constraints, e.g., a circular trajectory, the virtual mechanisms can be designed as in Fig. 1b. A spring and damper connect the end effector to a virtual L-shaped structure, which has mass and is free to rotate relative to the table, driven by additional spring and damper. The spring and damper parameters and the geometry and position of the virtual structure concur to determine the virtual force that drives the end effector towards a circular motion. Compared to Fig. 1a, Fig. 1b shows how virtual model control can be used for (compliant) constrained motion. The resulting virtual force is given by

$$\boldsymbol{f}_{EE} = k\boldsymbol{x} + c\dot{\boldsymbol{x}},\ x = g(\theta) - \boldsymbol{x}_{EE} \quad (3a)$$

$$I\ddot{\theta} + c_\theta \dot{\theta} + k_\theta(\theta) = w(\theta, \dot{\theta}, \boldsymbol{f}_{EE}), \quad (3b)$$

where θ is the angle from table of the virtual L-shaped part with inertia I, c_θ and k_θ are rotational damping and rotational

stiffness parameters, respectively. g represents the effect of the geometry of the L-shaped part. $w(\theta,\dot{\theta},f_{EE})$ is the torque on the L-shaped part induced by the robot.

To cover large distances, a 'moving' target approaches mitigates force exertion, as shown in Fig. 1c. That is,

$$\boldsymbol{f}_{EE} = k\boldsymbol{x} + c\dot{\boldsymbol{x}},\ \boldsymbol{x} = \boldsymbol{x}_d(t) - \boldsymbol{x}_{EE} \tag{4a}$$

$$\boldsymbol{x}_d(t) = \boldsymbol{x}_0 + s(t)(\boldsymbol{x}_g - \boldsymbol{x}_0). \tag{4b}$$

$\boldsymbol{x}_0$ is the initial position of the end effector. The straight motion of the moving target $\boldsymbol{x}_d(t)$ is regulated by the parameter $0 \leq s(t) \leq 1$, starting at zero and converging to one in finite time. The robot follows the moving target in a compliant way.

Finally, Fig. 1d offers a more effective control of the force that the end effector exerts at contact with the environment. A virtual cart is attached to the end effector through spring-damper components and is pulled with a predefined constant force along a straight trajectory. The virtual force on end effector depends on the cart's dynamics.

$$\boldsymbol{f}_{EE} = k\boldsymbol{x} + c\dot{\boldsymbol{x}},\ x = \boldsymbol{x}_C - \boldsymbol{x}_{EE} \tag{5a}$$

$$m\ddot{\boldsymbol{x}}_C + b\dot{\boldsymbol{x}}_C = \boldsymbol{f}_C + w(\boldsymbol{x}_C, \dot{\boldsymbol{x}}_C, \boldsymbol{f}_{EE}), \tag{5b}$$

where the second equation models the additional dynamics of the cart. $\boldsymbol{x}_C$ and $\dot{\boldsymbol{x}}_C$ are cart position and velocity. The cart has mass m, its motion is affected by viscous friction b, and it is pulled by the force $\boldsymbol{f}_C$. $\boldsymbol{w}(\boldsymbol{x}_C, \dot{\boldsymbol{x}}_C, \boldsymbol{f}_{EE})$ models the reaction force on the cart due to the robot.

In the presence of obstacles or obstructions that stop the motion of the end effector, the virtual mechanism (5) guarantees that the total force exerted by the robot is bounded by the cart's force $\boldsymbol{f}_C$. When the end effector is obstructed, also the cart reaches a stop. This means that the 'desired' motion is naturally paused and can be smoothly restored by removing the obstacle. These features show how the introduction of virtual model control is well adjusted to support interactions with a partially known environment.

C. Obstacle avoidance and compliant interaction

In its simplest form, obstacle avoidance is achieved by a repulsive virtual spring and a virtual damper that connect robot and obstacle, as shown in Figs. 1e and 1f. We consider nonlinear springs that become stiffer as their length reduces. Thus, in combination with other virtual mechanisms, this strategy allows to prioritize obstacle avoidance when the robot is near obstacles.

Focusing on the end effector for simplicity, the driving virtual force $\boldsymbol{f}_{\mathrm{o}}$ due to several repulsive springs attached between N obstacles and end effector is given by

$$\boldsymbol{f}_{\mathrm{o},i} = k_i(|\boldsymbol{x}_i|)\boldsymbol{x}_i + c_i(|\boldsymbol{x}_i|)\dot{\boldsymbol{x}}_i,\ \boldsymbol{x}_i = \boldsymbol{x}_{EE} - \boldsymbol{x}_{\mathrm{o},i}, \tag{6a}$$

$$\boldsymbol{f}_{\mathrm{o}} = \Sigma_{i=1}^{N} \boldsymbol{f}_{\mathrm{o},i}, \tag{6b}$$

where $\boldsymbol{x}_{\mathrm{o},i}$ is the position of the i-th obstacle. $k_i(|\boldsymbol{x}_i|)$ and $c_i(|\dot{\boldsymbol{x}}_i|_i)$ are the associated nonlinear stiffness and nonlinear damping, respectively.

The case of unknown or uncertain obstacles is more challenging. For the unknown interactions of cluttered environments we need to take advantage of the controlled compliance of the robot. Consider the scenario of an undetected obstacle (acrylic plate) shown in Fig. 9. The behavior of the robot is determined by the interaction of its mechanics, the virtual mechanism, and by the additional contact with the obstacle. In fact, this contact force adds up to the forces generated by the virtual mechanism, therefore we can design the virtual mechanisms to guarantee that the robot behavior will degrade in a physically predictable manner. In case of obstruction, for example, the static picture is that the end effector will stop at the balance between the force exerted by the virtual mechanism springs and the reaction force from the obstacle. The total force expressed by the robot is thus regulated by the stiffness of the virtual springs and by the mechanical properties of the obstacle. Indeed, a soft obstacle will have a different effect than a rigid material.

III. Robot Platform and Control Architecture

The experimental platform utilized for the study is the humanoid robot Sciurus17 shown in Fig. 2, a lightweight low cost platform developed by RT Corp. It has 17 degrees of freedom (DoF) distributed across its upper body, with seven DoF for each arm, two DoF for head and one DoF for torso. The robot uses Dynamixel motors capable of current control. The conversion from torque to current is approximated as a linear equation using the performance graph of the Dynamixel motor. The robot head is also equipped with an Intel RealSense D415 camera, which can be used to detect obstacles.

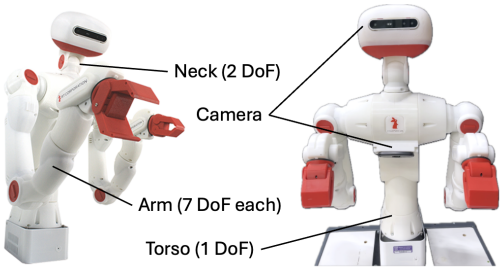


Fig. 2: Experimental platform: Sciurus17 robot.

The controller runs on a standard computer. The control architecture is depicted in Fig. 3. The virtual forces generated by the virtual mechanisms are computed and translated into joint torques using (1). These are then converted to demanded currents using the approximated linear relationship. The computation of the 'state' of virtual mechanisms, such as the elongation of virtual springs and the velocities for the dampers, relies on position and velocity information of the robot joints, read from the Dynamixel motors. Inverse kinematics is not needed to control the robot. All forces and their translation into joint torques depend on forward kinematics and associated Jacobians. The calculation involves only adding and multiplying matrices, as shown in (1). This

is a significant departure from classical motion planning, whose execution often relies on inverse kinematics of the robot.

For obstacle detection, vision information such as image or point clouds can be captured from the camera. Once the obstacle is segmented, the nearest point on its bounding box or point cloud to the end effector can be easily identified. This point can then be used as the obstacle position in (6a). However, for the experiments of this paper, vision-based obstacle detection is not used. The position and radius of the obstacles are explicitly encoded.

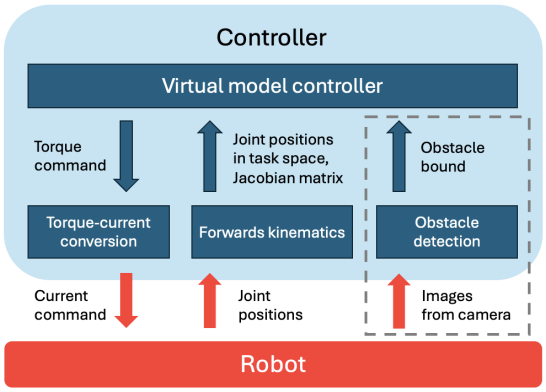


Fig. 3: Control architecture.

IV. Experiments

A. Reaching in obstacle-free environment

Two reach motions are compared in Fig. 4, both generated using *simple reaching* (2), for spring stiffness $k = 15$ N/m and $k = 45$ N/m while maintaining a constant damping coefficient of 1.5 Ns/m. It is not difficult to predict the behaviors for $k = 15$ and $k = 45$. The latter attains a smaller error and faster motion as shown in Fig. 4 (bottom). In fact, for $k = 15$ N/m, the force of the spring in the last section of the black trajectory is not enough to overcome friction. During transients, the end effector tends to deviate along the y-axis before reaching the goal. This is the result of the interaction between the virtual spring, the robot's arm inertia, and the inaccuracy of torque-current conversion for high-speed movements.

Fig. 5 shows the trajectories obtained by *moving-reference reaching* (4). Trajectories are generated using spring stiffness $k = 45$ N/m and damping coefficient of 1.5 Ns/m. The moving reference has a trapezoidal-shaped speed profile. Compared to Fig. 4, moving-reference reaching limits high forces and fast motions. The deviation along the y-axis is reduced. In fact, at lower speeds, the impact of imprecise torque-current conversion and robot inertia is diminished. As in the previous case, the 5 cm error from the target is due to the low stiffness of the reach motion. The force applied at the end of the trajectory does not overcome friction.

B. Reaching while avoiding known obstacles

A cylindrical obstacle with 10 cm diameter is placed as shown in Fig. 6 (top). The black line shows the trajectory

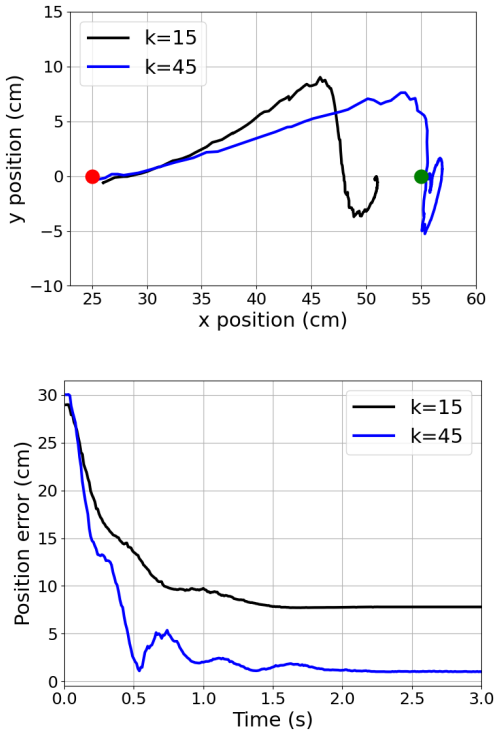


Fig. 4: **Simple reaching with different stiffness**. Black line: $k = 15$. Blue line: $k = 45$. Red dot: starting position. Green dot: goal. Top: trajectory of the end effector on the x-y plane. Bottom: time-evolution of the reach error.

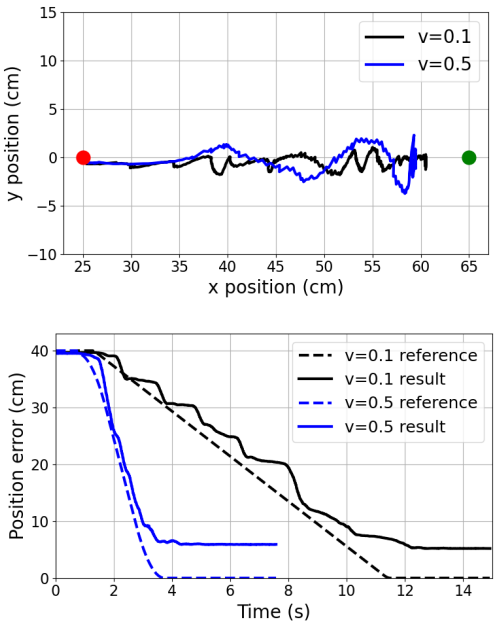


Fig. 5: **Moving-reference reaching with different speeds**. Black line: 0.1 m/s. Blue line: 0.5 m/s. Red dot: starting position. Green dot: goal. Top: trajectory of the end effector on the x-y plane. Bottom: time-evolution of reference and actual distances from the target.

of the end effector in the x-y plane. The robot is driven by moving-target reaching (4), with speed 0.1 m/s, combined to *obstacle avoidance* (6). That is, for the latter, the obstacle is avoided through the action of a repulsive virtual nonlinear spring between the end effector and the obstacle. The stiffness of the virtual spring is given by (in N/m)

$$\begin{aligned} 0 \le r < 5\,\text{cm}, &\quad k = -46r + 250 \\ 5\,\text{cm} \le r < 10\,\text{cm}, &\quad k = -3.6r + 38 \\ 10\,\text{cm} \le r < 20\,\text{cm}, &\quad k = -0.2r + 4 \\ r \ge 20\,\text{cm}, &\quad k = 0 \end{aligned}$$

where r is the distance from end effector to the closest point on the obstacle.

The virtual model control offers a simple interpretation of the behavior of the robot. The end effector initially follows compliantly the straight-line reference. It is then pushed away as it gets closer to the obstacle, the motion being determined by the balance of forces between the repulsive spring from the obstacle and the attractive spring from the reference. Finally, after getting rid of the obstacle, the end effector accelerates towards the goal.

Two cylindrical obstacles with 6 cm diameter are then placed as shown in Fig. 6 (middle). In this case, the obstacles are avoided by placing a repulsive virtual spring between end effector and each obstacle. It is not difficult to predict what the robot will do, as its motion is now determined by the combined action of two repulsive springs. In contrast to path planning algorithms, the presence of two obstacles does not cause a significant increase in computation, as no pre-planning or re-planning are required.

As the last scenario, consider the obstacles of Fig. 6 (bottom). These partially overlap, forming a nonconvex shape. The resulting trajectories are not surprising. The trajectories manage to avoid the obstacles after fluctuating for a while near the local minima of the elastic potential field formed by the virtual components. Similar to path planning with potential fields, this scenario shows how the combined effect of several springs may lead to issues. Indeed, this scenario calls for an improved design of the virtual model controller.

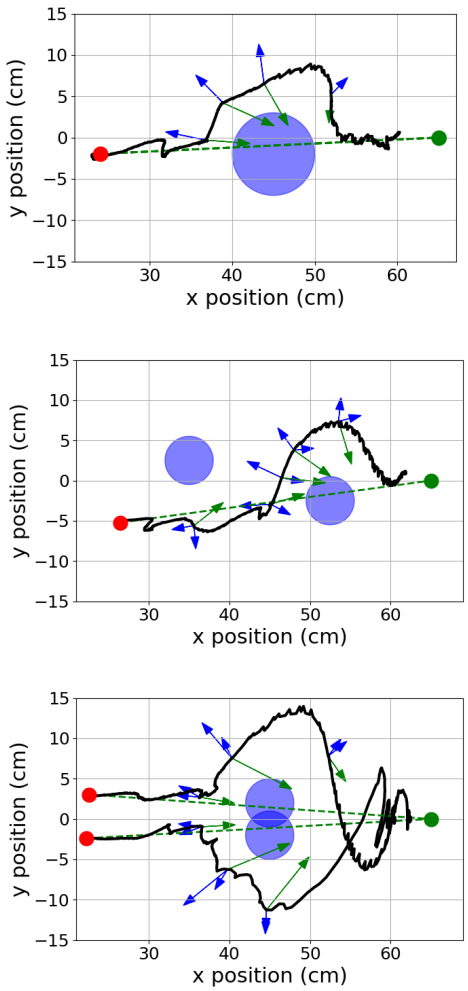


Fig. 6: **Obstacle and multi-obstacle avoidance with known obstacles**. Black line: robot trajectory. Blue area: cylindrical obstacles. Red dot: starting position. Green dots: ending positions. Green line: reference trajectory, 0.1 m/s. The arrows show the virtual mechanism forces. Blue arrow: force generated by obstacle avoidance virtual model. Green arrow: force generated by moving-reference reaching virtual model.

C. Reaching while interacting with unknown obstacles

The experiments of the previous sections show how different virtual components can be combined to achieve complex goals. Both sections take advantage of available information to drive the robot behavior. In contrast, in this section we look into uncertain interactions, given by undetected or inaccurately positioned obstacles.

As the first scenario, consider the reach task in Fig. 4, based on the simple reaching virtual model controller (2). This time, the end effector is blocked and held by a human hand in the middle of reaching, around $x = 40$ cm. The robot has to face an unexpected (undetected) obstruction, as shown in Fig. 7 (top). The reaction of the robot is the one we expect from the physical interpretation of the virtual model controller. The end effector temporarily stops. The force must be proportional to the stiffness of the spring and the distance to the target. The motion of the robot resumes as soon as the end effector is released. The interaction with the human hand has no effect on the final error. For comparison, we have also implemented PID control in joint space based on inverse kinematics (IK). The differences with IK are straightforward. The virtual model controller applies up to 4.53 N force compared to 35.77 N of the IK approach. Contact forces are reported in Fig. 8.

As the second scenario, consider the reaching task of Figure 5, based on the moving-reference reaching virtual model controller (4). The reference speed is 0.5 m/s. This time, an acrylic plate is placed within the robot operative space and partially obstructs the motion, as shown in Fig. 9. The presence and position of the plate are both unknown to the robot.

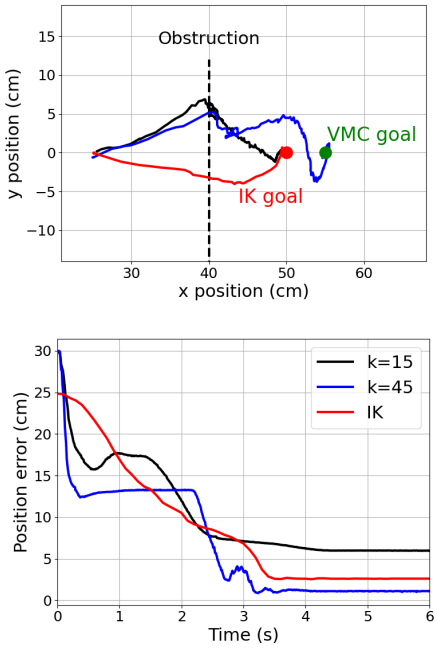


Fig. 7: **Simple reaching under undetected human interaction**. The end effector is blocked by a person at $x = 40$ cm. Black line: (2), $k = 15$. Blue line: (2), $k = 45$. Red line: inverse kinematics. Green dot: goal for VMC. Red dot: goal for inverse kinematics (which is different from VMC due to its inability to reach further using torso). Top: trajectory of the end effector on the x-y plane. Bottom: time-evolution of the reach error.

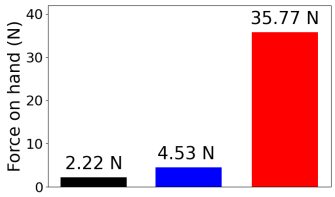


Fig. 8: **Force exerted by end effector during obstruction**. Black: (2), $k = 15$. Blue: (2), $k = 45$. Red: inverse kinematics.

Snapshots of the experiment and the resulting end effector trajectory are presented in Fig. 9. After colliding with the obstacle, the end effector is held in place by the obstacle for around 0.5 seconds, then slides along it, and finally reaches the goal. Virtual model control offers a simple explanation for this behavior. Once in contact with the obstacle, the end effector is held in place until the virtual spring between the end effector and the moving reference extends enough to exert the required force to overcome friction and slide along the acrylic plate. The goal position is eventually reached once the obstacle is cleared. Notably, the 'physical' perturbation of the acrylic plate shows similarities with the 'virtual' perturbation of the obstacles of Fig. 6.

V. Discussion and Conclusions

After its introduction by [4] for the control of walking bipedal robots, virtual model control remained underexplored

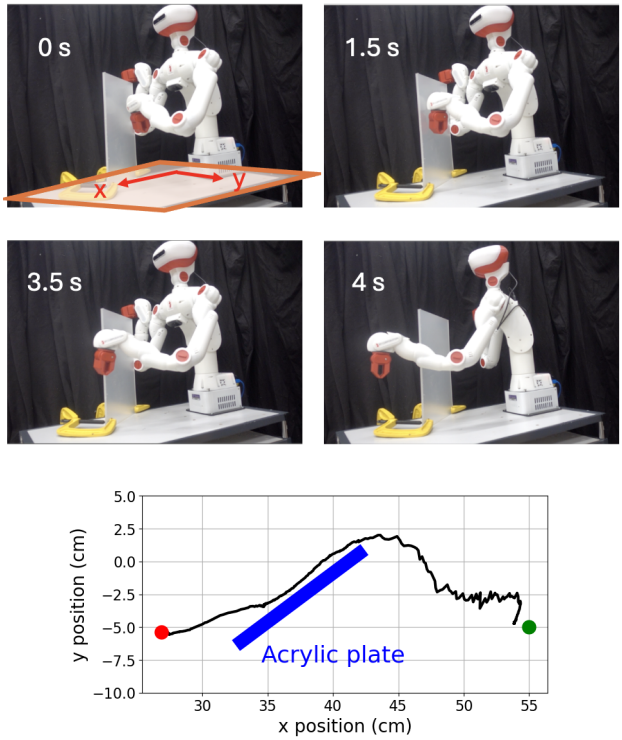


Fig. 9: **Moving-reference reaching under undetected obstacle**. Top snapshots: 0s - contact established, 1.5s - sliding, 3.5s - obstacle cleared, 4s - goal reached. Bottom figure: perturbed end effector trajectory. Black - trajectory, red dot - starting position, green dot - goal, blue - acrylic plate.

in robot manipulation, even if several contributions in the literature show significant similarities. In [15] virtual springs and dampers are used to control a robot into sequential patterns. Clear contact points can be found in the impedance control approach in end-effector space proposed by [16], [5]. In [17] we see how virtual components are used to control the position of a physically simulated character's hand, posture, and walking cycles. Although these contributions recognize the value of using virtual components such as springs and dampers for control design, they do not venture into exploring the complexity and flexibility of virtual model control, with the introduction of additional dynamic and complex mechanical structures. This path is taken by [18], [19] to design a new controller based on a 'virtual instrument' for robot laparoscopy. To the best of our knowledge, using virtual mechanisms for robot planning has not been explored in previous publications.

The examples of Section II and the experiments of Section IV support our hypothesis. Virtual model control appears to have a significant advantage in terms of practical and intuitive design. We design a virtual mechanisms for each task. Then, the robot can be controlled by activating the right collection of virtual mechanisms at run-time, as shown in the experiments. Section IV-B shows that the controlled robot is predictable. The motion of the robot aligns with the expected

interaction between the end effector, the environment, and the virtual mechanism. The benefits of robot compliance are also evident. The interaction with the unknown obstacle of Section IV-C bears minor consequences on the reach motion. The reach motion is temporarily halted when a person grabs the robot end effector, but is restored right after. In comparison with classical inverse kinematics we see how interaction forces are significantly reduced, enabling safer interactions.

In virtual model control, virtual components, such as springs and dampers, respectively, shape the potential energy of the controlled robot and the dissipation of the total energy. In this sense, for fully actuated robots, virtual model control provides a simple way to do energy-shaping and damping-injection [13], [7], [12]. The crucial difference is the design process. Representing the controller as a virtual mechanism allows to tailor the control action to the features of the task. In fact, consider the task of deriving an energy-shaping controller for reaching and avoiding obstacles. Finding the mathematical formula for the potential energy is a challenging task. In contrast, virtual model control guides the derivation of a desired potential energy, which is indirectly obtained as the combined energy of robot and virtual mechanism.

Interconnecting the robot with a compliant virtual mechanism is also a way to shape the closed-loop impedance of the robot [8], [20], [21], [5], [22]. In fact, a virtual mechanism with high stiffness and high damping coefficients makes the robot stiff, thus insensitive to external forces. In contrast, low stiffness and low damping coefficients make the robot motion very compliant, thus sensitive to the interaction with the environment. Although impedance control is a well-established and powerful methodology, finding the 'right' impedance for a given task remains an elusive problem. In practice, impedance control too often reduces to the combination of dynamics inversion (to remove nonlinearities) and simple linear impedances. In contrast, virtual model control enables the design of complex nonlinear impedances, tailored to the task requirements.

The purpose of this paper is to illustrate some fundamental features of virtual model control in dealing with integrated motion control and motion planning. Virtual model control provides an intuitive design approach for controlled robots that interact with the uncertain environment in a predictable way. Its design flexibility lies in the use of nonlinear mechanical elements organized in complex mechanical structures. Thus, virtual model control can take advantage of impedance compositionality, passivity, and additional compliant (virtual) dynamics to deliver robust and safe control for robots operating in unstructured and dynamic environments. That said, we are aware that this paper only scratches the surface of virtual model control design.

Much remains to be understood. The choice of specific virtual mechanisms and the tuning of the virtual mechanical parameters were based on a trial-and-error approach. Optimal mechanisms and optimal parameter tuning are crucial research directions to address, with some initial results in [19]. In a similar way, task prioritization in complex settings, the trade-off between precision and compliance, and the use of adaptive virtual mechanisms to optimize interactions at run time are key research directions to bring virtual model control to a mature design approach.

REFERENCES

[1] S. M. LaValle, "Rapidly-exploring random trees: a new tool for path planning," *The annual research report*, 1998.

[2] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[3] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.

[4] J. E. Pratt, C.-M. Chew, A. Torres, P. Dilworth, and G. A. Pratt, "Virtual model control: An intuitive approach for bipedal locomotion," *The Int. Journal of Robotics Research*, vol. 20, pp. 129 – 143, 2001.

[5] F. Ficuciello, L. Villani, and B. Siciliano, "Variable impedance control of redundant manipulators for intuitive human–robot physical interaction," *IEEE Transactions on Robotics*, vol. 31, no. 4, pp. 850–863, 2015.

[6] C. Lei, Z. Zhong, J. He, and J. Zhang, "A method of manipulator dragging teaching based on VMC," in *International Conference on Computer, Artificial Intelligence, and Control Engineering (CAICE 2023)*, ser. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, X. Feng and A. Bhattacharjya, Eds., vol. 12645, May 2023, p. 1264509.

[7] R. Ortega, A. Van Der Schaft, I. Mareels, and B. Maschke, "Putting energy back in control," *Control Systems, IEEE*, vol. 21, no. 2, pp. 18 –33, 4 2001.

[8] N. Hogan, "Impedance Control: An Approach to Manipulation: Part I—Theory," *Journal of Dynamic Systems, Measurement, and Control*, vol. 107, no. 1, pp. 1–7, 03 1985.

[9] M. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. Wiley, 2005.

[10] J. Willems, "Dissipative dynamical systems part I: General theory," *Arc. for Rational Mechanics and Analysis*, vol. 45, pp. 321–351, 1972.

[11] A. van der Schaft, *L_2-Gain and Passivity in Nonlinear Control*, 2nd ed. Secaucus, N.J., USA: Springer-Verlag New York, Inc., 1999.

[12] R. Ortega, J. Romero, P. Borja, and A. Donaire, *PID Passivity-Based Control of Nonlinear Systems with Applications*. Wiley, 2021.

[13] R. Ortega, A. Loría, P. Nicklasson, and H. Sira-Ramírez, *Passivity-based Control of Euler-Lagrange Systems: Mechanical, Electrical and Electromechanical Applications*, ser. Communications and Control Engineering. Springer London, 1998.

[14] J. Colgate and N. Hogan, "Robust control of dynamically interacting systems," *Int. Journal of Control*, vol. 48, no. 1, pp. 65–88, 1988.

[15] R. Brooks, L. Aryananda, A. Edsinger, P. Fitzpatrick, C. Kemp, U.-M. O'Reilly, E. Torres-Jara, P. Varshavskaya, and J. Weber, "Sensing and manipulating built-for-human environments," *International Journal of Humanoid Robotics*, vol. 01, no. 01, 2003.

[16] T. Senoo, M. Koike, K. Murakami, and M. Ishikawa, "Impedance control design based on plastic deformation for a robotic arm," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 209–216, 2017.

[17] R. Backman and M. Kallmann, "Modeling physically simulated characters with motion networks," in *Motion in Games*. Springer Berlin Heidelberg, 2012, pp. 78–89.

[18] D. Larby and F. Forni, "A generalized approach to impedance control design for robotic minimally invasive surgery," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 8548–8555, 2023, 22nd IFAC World Congress.

[19] ——, "A passivity preserving h-infinity synthesis technique for robot control," in *2022 IEEE 61st Conference on Decision and Control (CDC)*, 2022, pp. 1416–1422.

[20] N. Hogan, "Impedance Control: An Approach to Manipulation: Part II—Implementation," *Journal of Dynamic Systems, Measurement, and Control*, vol. 107, no. 1, pp. 8–16, 03 1985.

[21] ——, "Impedance Control: An Approach to Manipulation: Part III—Applications," *Journal of Dynamic Systems, Measurement, and Control*, vol. 107, no. 1, pp. 17–24, 03 1985.

[22] ——, "Contact and physical interaction," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, no. 1, pp. 179–203, 2022.