

Catching Flying Balls and Preparing Coffee: Humanoid Rollin'Justin Performs Dynamic and Sensitive Tasks

Berthold Bäuml, Florian Schmidt, Thomas Wimböck, Oliver Birbach, Alexander Dietrich, Matthias Fuchs, Werner Friedl, Udo Frese, Christoph Borst, Markus Grebenstein, Oliver Eiberger, and Gerd Hirzinger

Abstract—The mobile humanoid Rollin'Justin is a versatile experimental platform for research in manipulation tasks. Previously, different state of the art control methods and first autonomous task execution scenarios have been demonstrated. In this video two new applications with challenging task requirements are presented. One is the catching of one or even two flying balls using all of Justin's degrees of freedom. The other is the autonomous preparation of coffee. Both applications need adequate sensors to support local referencing. The required precision in position and timing is realized in software, using the sensor information, taking the varying precision of Justin's kinematic sub-chains into account and handling all timings in sub-millisecond range.

I. INTRODUCTION

The mobile humanoid system Rollin'Justin is designed as a platform for research on two handed manipulation and autonomous task execution in everyday human environments. Justin consists of two lightweight robot arms (2x7DOF) [1], a torso with head (5DOF), two hands (2x12DOF) [2], and an omnidirectional base [3]. The whole upper body is constructed following lightweight principles. Each of the actuated joints has a built-in link side torque sensor. This allows to realize compliant control features, which have been already applied to different demonstration scenarios [4].

The lightweight construction of the system, however, leads to non-negligible elasticities in the system. As far as they result from the joint design they are compensated using a flexible joint model in the control loop. This reduces the robot's tracking error - comparable to very stiff industrial robots. Flexibility of the structure and in the tendon mechanism of the torso, however, cannot be compensated directly. The calibration of a control model to account for these errors is complex and is in practice not able to completely cancel the effects. This work shows how these errors, which will be much more challenging in the new generation of intrinsically elastic robots (e.g. [5], [6]) as well as other uncertainties, e.g. in the visual perception system or due to slippage of the wheel esp. for dynamic movements, can be intelligently compensated in the higher level software layers of Justin. The key issue is to define the right local reference systems using sensor information combined with taking the various precisions of the kinematic sub-chains into account. To allow

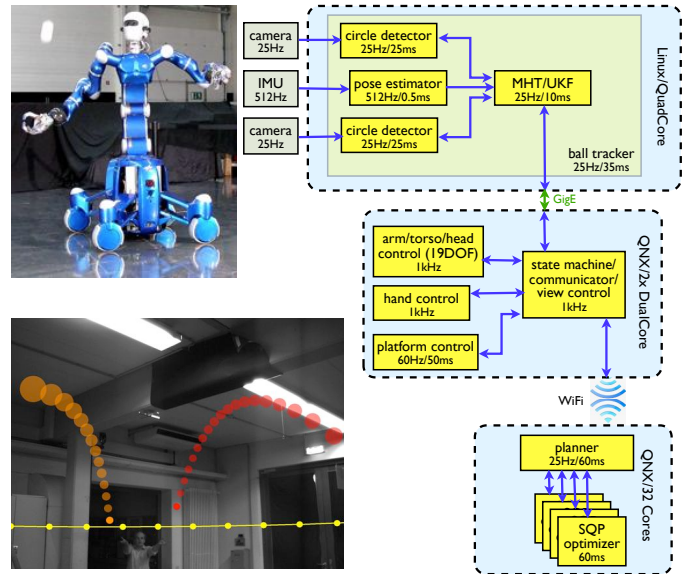


Fig. 1. How Rollin'Justin catches flying balls. Stereocameras track the thrown balls and an inertial measurement unit determines the 6DOF head pose (see yellow artificial horizon in camera view) [7] to account for not completely cancelable vibrations introduced to the robot's light weight structure due to the rapid catch motions. Based on the continuously improving trajectory predictions a realtime path planner, running on an external cluster with 32 cores, decides where, when, and in which configuration to kinematical optimally catch the ball by repeatedly solving a nonlinear optimization problem including simple collision avoidance. In [8] we described a previous version of the planner for only one arm and simpler collision avoidance geometries. The resulting joint paths are executed by the arms, torso and mobile platform. The head joints are controlled to keep the balls in the cameras' field of view. All software components communicate in hard realtime by means of the institute's arD (agile Robot Development) software framework [9].

this scheme also for highly dynamic tasks¹, in addition the software has to thoroughly keep track of all timings in the sub-millisecond range.

Two applications are presented illustrating the level of versatility the humanoid Justin has reached by this pairing of a high-performance hardware with "intelligent" higher level software: (1) The highly dynamic task of catching flying balls thrown towards the robot. (2) Preparing a cup of coffee with a pad coffee machine, requiring a complex sequence of fine manipulation operations with high precision of the fingertips relative to the objects.

¹i.e. tasks requiring the (flexible joint) robot to move in a regime, where inertial forces and effects due to elasticity get important.

II. CATCHING FLYING BALLS

Catching a thrown ball with a hand is not easy. A tight interplay of fast perception, a good catching strategy, body control, and dexterity is needed. Because of this, ball catching has been used for almost 20 years now as a challenging benchmark system to develop and test robotics key technologies [10], [11], [12], [13], [8].

The presented demonstrator (Fig. 1) reaches a new level: beside the high number of DOFs (including the torso, head and mobile platform) involved in the catching motion and the catching of two balls at the same time, all sensing, esp. the camera system, is *onboard* and the whole system operates completely wireless. The system robustly reaches the necessary precision² of 2cm in space and 5ms in time leading to a catch rate³ of over 80%.

III. PREPARING COFFEE

Preparing a cup of coffee (Fig. 2) requires precise fine manipulation with the fingertips. It is difficult to achieve the necessary precision in localizing objects and execution of the task by using only a single visual world model update and then to grasp blindly. Therefore, a new execution environment for Justin has been built combining vision and force sensing (Fig. 3).

The manipulation of the small coffee pad and the fragile paper cup for example requires high precision relative to the object and sensitivity not to damage the cup. To precisely grasp the pad a guarded move monitoring contact to the supporting surface is executed before the fingers are closed.

Another challenge is the insertion of the pad into the coffee machine. The opening and closing of the pad drawer needs precision in the millimeter range. The hand is moved to the machine monitoring the external finger forces till a certain contact situation around the drawer handle is reached. However, in most cases there is still a remaining position error of the hand relative to the coffee machine. In normal grasp operations this error is balanced by a moderate stiffness of the fingers. Using such a strategy here often ends with an unintended slipping of the handle and so loosing the drawer. Therefore, in this case a high grasp stiffness is commanded in order to exploit self-alignment effects. In return to limit the interaction forces due to remaining position errors a moderate stiffness of the arm tool center point is set allowing to pull out the drawer robustly. A special operation is introduced that ensures that the drawer can be reinserted robustly by computing appropriate command offsets considering a flexible joint robot model.

REFERENCES

- [1] G. Hirzinger, N. Sporer, and et.al., “DLR’s torque-controlled light weight robot III - are we reaching the technological limits now?” in *ICRA*, 2002, pp. 1710–1716.

²defined by the hand geometry and the ball size (8cm diameter), so that the ball does not hit the fingers of the open hand and does not bounce out of the hand again before the fingers can close.

³percentage of successful catches for typical throws (distance of ca. 6m) into the feasible *catch space*, i.e. the workspace the robot can reach during the ball flying time (0.6 to 1.0s).

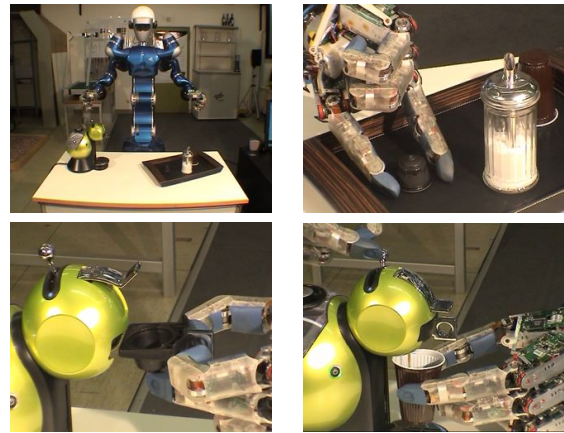


Fig. 2. The presented coffee preparation scenario. A pad is grasped, inserted into the machine, the cup is taken, and the machine is turned on and off.

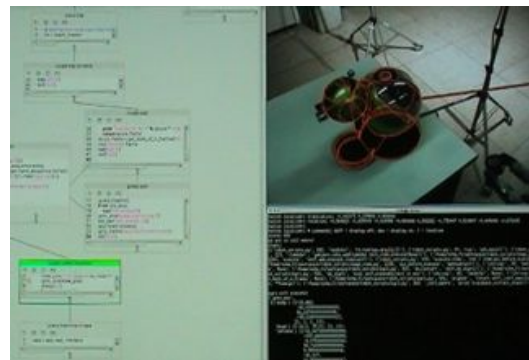


Fig. 3. The new scripting interface allows to define complex control flows connecting python script building blocks. These scripts trigger elementary operations, running in the realtime control flow. To ensure immediate reaction (realtime) guards can be defined to monitor the control execution. For many tasks, this allows the online generation of control code without considering particular realtime software issues.

- [2] J. Butterfaß, M. Grebenstein, H. Liu, and G. Hirzinger, “DLR-Hand II: Next generation of a dextrous robot hand,” in *ICRA*, 2001.
- [3] M. Fuchs, C. Borst, and et.al., “Rollin’ justin - design considerations and realization of a mobile platform for a humanoid upper body,” in *ICRA*, 2009, pp. 4131–4137.
- [4] C. Borst, T. Wimböck, and et.al., “Rollin’ justin: Mobile platform with variable base,” in *ICRA*, 2009, pp. 1597–1598.
- [5] M. Grebenstein, A. Albu-Schäffer, and et. al., “The dlr hand arm system,” *ICRA*, 2011.
- [6] T. Sugaiwa, H. Iwata, and S. Sugano, “New visco-elastic mechanism design for flexible joint manipulator,” in *AIM*, 2008.
- [7] O. Birbach, U. Frese, and B. Bäuml, “Realtime perception for catching a flying ball with a mobile humanoid,” in *ICRA*, 2011.
- [8] B. Bäuml, T. Wimböck, and G. Hirzinger, “Kinematically optimal catching a flying ball with a hand-arm-system,” in *IROS*, 2010.
- [9] B. Bäuml and G. Hirzinger, “When hard realtime matters: Software for complex mechatronic systems,” *Robotics and Autonomous Systems*, vol. 56, no. 1, pp. 5–13, 2008.
- [10] B. Hove and J. Slotine, “Experiments in robotic catching,” in *Proceedings of the American Control Conference*, 1991, pp. 380–385.
- [11] K. Nishiwaki, A. Konno, and et.al., “The humanoid saika that catches a thrown ball,” in *IEEE Workshop on Robot and Human Com.*, 1997.
- [12] U. Frese, B. Bäuml, and et.al., “Off-the-shelf vision for a robotic ball catcher,” in *IROS*, 2001.
- [13] M. Riley and C. G. Atkeson, “Robot catching: Towards engaging human-humanoid interaction,” *Autonomous Robots*, vol. 12, pp. 119–128, 2002.