2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

Stabilizing Hybrid Switched Motion Control Systems with an On-Line Trajectory Generator

Torsten Kröger and Friedrich M. Wahl

***Abstract*— This paper suggests the *idea* of a universal method for stabilizing discrete-time hybrid switched-control systems of robot manipulators. The core of this idea is based on an *on-line trajectory generation algorithm* that is able to generate continuous command variables from any arbitrary state of motion. We define a measurable criterion to on-line detect an instability or a potential instability of the plant, and right after this criterion is fulfilled, we switch to the on-line trajectory generator that acts as an open-loop pose control submodule in the switched-system. The on-line trajectory generation algorithm guides the system under consideration of kinematic motion constraints to a desired target state of motion that can be specified beforehand (e.g., zero-velocity in a pre-defined position). Systems with one and more degrees of freedom are regarded in this paper; finally, real-world experimental results achieved with a six-joint industrial manipulator are presented in order to demonstrate the potential and the high practical relevance of this concept.**

I. Introduction

In order to achieve future advancements in the field of robot motion control, the integration of sensors and sensor-based control is indispensable. Sensor-based control in this context does not only mean the calculate motion trajectories based on measured sensor-signals but to embed sensors in the feedback loops, such that, for example, force/torque, distance, and/or visual servo control becomes feasible. In practice — of course — these control modes have to be combined with common trajectory-following motions. It is of enormous practical relevance to switch from trajectory-following motion control to closed-loop sensor-based motion control and vice versa at unforeseen instants. Hybrid switched-control systems are systems comprised of a family of continuously working subsystems and a supervisory discrete system that switches between them. Switching may be initiated by time, by state, and or by sensor signals in order to execute a specified task; all of them may happen at unforeseen instants, and the control system has to switch instantaneously from one control cycle to another.

Proving stability of hybrid switched-control systems is a well-known and demanding task, and generic solutions are only available for concrete system classes. Instabilities of such systems can be caused by several reasons: wrong/incomplete task specification, inappropriate switching sequences, incorrectly chosen control submodules, unexpected environments, or sensor failures.

In this paper, we assume a motion control system with one or more degrees of freedom (DOF) equipped with one or more sensors delivering digital and/or analog sensor signals; each sensor may be part of the feedback control loop. Fig. 1 shows an abstract and simple scheme of a hybrid switched-system control architecture for a one-DOF system, that is, with one actuator only (e.g., a simple linear positioning unit). The inner control loops (i.e., behind the switched signal) are assumed to be Lyapunov stable. The basic idea is: If the inner control system receives command variables from one of the closed-loop controllers, and the overall system tends to become instable, the supervisory switching unit selects either the open-loop position or the open-loop velocity controller to provide input signals for the inner loops. These two submodules are on-line trajectory generation (OTG) algorithms; their usage for stabilizing hybrid switched-motion control systems constitutes the core part of this paper.

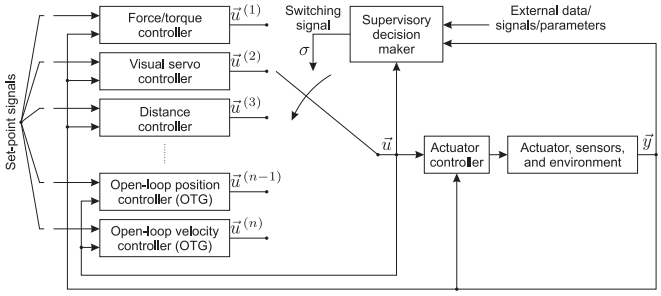

Fig. 1. Abstract control scheme of a hybrid switched-system for one single DOF, that is, one single actuator. The usage of the two on-line trajectory generation (OTG) submodules for stabilizing such control systems in multiple DOFs constitutes the major contribution of this paper.

Before briefly introducing the basics of OTG algorithms for one- and multi-DOF systems in Sec. III, we give an overview about related works in the next section. Sec. IV presents the stabilization method, and Sec. V shows real-world experimental results achieved with industrial six-joint manipulators highlighting the relevance of this method.

II. Related Works

This section discusses works on hybrid switched-system control and stability analysis of such systems, which are tightly related to this contribution.

The basic requirement for overall stability of a motion control system as depicted in Fig. 1 is a proof of stability for the control system in joint/actuator space. This problem was surveyed very deeply, and we can find plenty of works and textbooks on this topic (e.g., [1]–[6]). Designing a robot joint control scheme is, of course, not a simple task, and there is *no golden rule* for it; commonly, a tradeoff between

T. Kröger and F. M. Wahl are with the Institut für Robotik und Prozessinformatik at the Technische Universität Carolo-Wilhelmina zu Braunschweig, Mühlenpfordtstraße 23, D-38106 Braunschweig, Germany {t.kroeger|f.wahl}@tu-bs.de.

performance and robustness is required, but the treatment of this topic clearly goes beyond the scope of this paper.

If we consider the actuator space control scheme as Lyapunov stable [7], we can focus on the task space control scheme, which contains the hybrid switched-system, and the analysis of such systems is one of the fundamental requirements of the work presented here. Especially, the works of Branicky [8], [9] and Liberzon [10], [11] provide elementary concepts to develop and analyze hybrid switched-system control techniques. In particular, the stability analysis is of fundamental interest here, because the stability of a switched-system cannot be assured by the stability of each single sub-controller. To prove the stability of hybrid switched-systems can be extremely difficult and many researchers are working on analyzing such stability questions. Brockett [12] explains this subject for motion control systems. Žefran and Burdick [13], [14] suggest an approach, whose purpose is very similar to the one of this paper; a system with changing dynamics is considered and a hybrid controller is designed for handling the system in different regimes" of dynamics.

It may happen that a set of stable subsystems becomes unstable if the switching between them occurs inappropriately [11], [15]–[17]. In the field of stability analysis, we can distinguish between techniques for linear [11], [18], [19] and nonlinear [7], [17], [20]–[22] switching systems. If all control submodules behave stably, the system can become unstable due to inappropriate switching sequences. The literature on such systems commonly distinguishes hybrid switched-systems into three subgroups [10]:

- State-dependent switching systems
- time-dependent switching systems, and
- autonomously switching systems.

The system to be regarded here (cf. Fig. 1) is an autonomously switching one: The switchings occur in dependence on the set-point sets and the external data. For the field of robot manipulation control, where we have to control a nonlinear plant, these set-points may be specified by Manipulation Primitives [23]. Till now, there is no method that exists to prove the stability of this class of systems. For concrete set-ups, concrete environments, and concrete input parameters, it is possible to apply multiple Lyapunov functions and Lie-algebraic stability criteria — but this is not a general solution [8]–[10], [15]. Even in this paper, we do not provide a formal stability proof but only discuss a logical line of arguments that leads to a heuristic and plausible result.

In this context and for the field of hybrid switched-control of robot manipulators, we consider force/torque control submodules [24]–[27], visual servo control submodules [28], and distance control submodules as closed-loop control submodules, whereas the realization of the latter ones is supposed to be a trivial task. A hybrid control approach for visual servoing applications was presented by Gans et al. [29]–[31]; here two visual servo controllers were suggested as submodules in a hybrid switched-system. Assuming an eye-in-hand camera setup, the first control module uses the camera position to calculate an error signal in the feedback loop of the control law and the second submodule uses image features. Stability is proven by means of a state-based switching scheme. Kühnlenz and Buss [32] present a Lyapunov-based stability proof for system that switches between several cameras.

The stabilizing method proposed in this paper is based on a class of OTG algorithms that was introduced in [33], [34]. How these algorithms work and what their properties are is briefly explained in the next section. If used as control submodules (cf. Fig. 1), Sec. IV suggests these algorithms to stabilize hybrid switched-control systems after an instability or a potential instability was detected.

III. Background: On-Line Trajectory Generation

An on-line trajectory generation algorithm can be considered as an open-loop position or pose controller using the current state of motion for command variable generation [33], [34]. Depending on the current state of motion, a motion profile is selected from a finite set of profiles (i.e., Types I,II use velocity profiles, Types III – V use acceleration profiles, and Types VI – IX use jerk profiles). Based on this profile a system of nonlinear equations can be set up, whose solution contains all trajectory parameters to transfer the system from its current state of motion to a desired target state of motion while considering given kinematic motion constraints. Each system of equations features a proper input domain, for which a valid solution can be found. For this class of algorithms, it is essential that the *union of all input domains equals the entire input domain* of the algorithm. In the following two subsections, we briefly describe the algorithms for open-loop velocity control and open-loop position (pose) control.

A. Open-Loop Velocity Control

The input and output values of the velocity-based algorithm for the OTG Types III – V (cf. [33], [34]) are depicted in Fig. 2. The current state of motion of one single DOF k at instant T_i is represented by the position ${}_kP_i$, the velocity ${}_kV_i$, and the acceleration ${}_kA_i$; the kinematic motion constraints for jerk limited trajectories are the maximum acceleration value ${}_kA_i^{max}$ and the maximum jerk value ${}_kJ_i^{max}$. It is the purpose of the algorithm to transfer this DOF k to the target

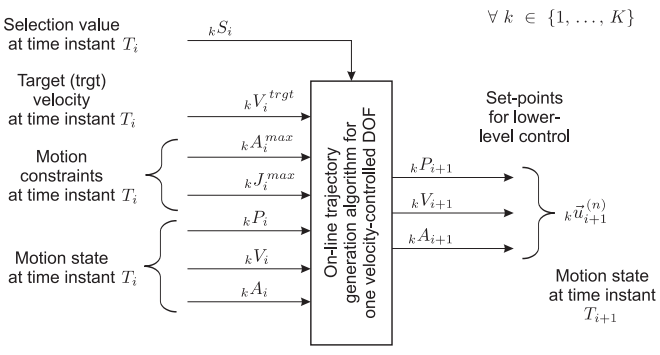

Fig. 2. Input and output values of the OTG algorithm for one single velocity-controlled DOF k (Types III – V, cf. [33], [34]; submodule n in Fig. 1).

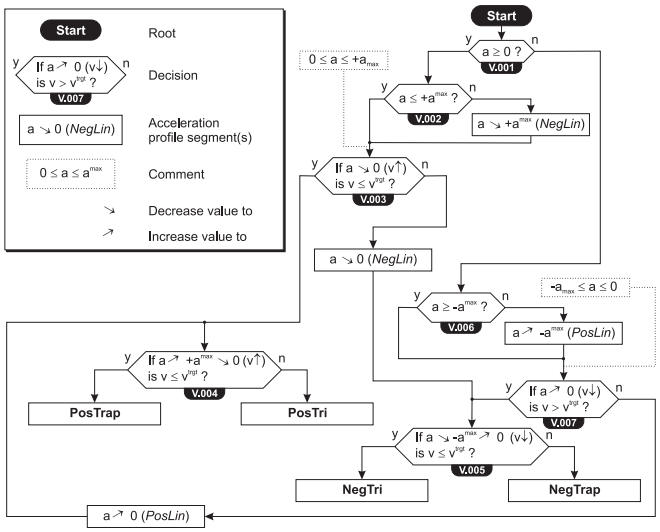

Fig. 3. Complete decision tree of the on-line trajectory generation algorithm for velocity-controlled DOFs. It is required for the control submodule n in Fig. 1, which is furthermore detailed in Fig. 2.

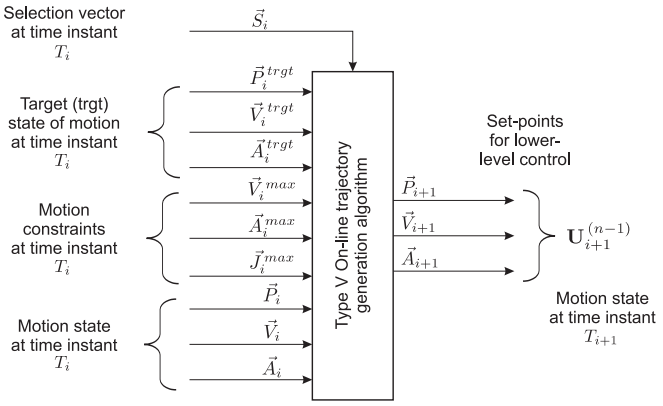

Fig. 4. Input and output values of the Type V OTG algorithm for multiple DOFs (cf. [33], [34]; submodule $(n-1)$ in Fig. 1).

velocity ${}_kV_i^{trgt}$ within the shortest possible time, that is, time-optimally. Assuming, we start at T_0, this state is reached after N control cycles:

$$ {}_kV_i \;=\; {}_kV_i^{trgt} \;\wedge\; {}_kA_i \;=\; 0 \;\forall\; i \;\in\; \{q \;\in\; \mathbb{N} \,|\, q \;\geq\; N\} \;. \quad (1) $$

Hence, the input domain for one single DOF is $\mathbb{R}^6$, and the algorithm has to be capable to compute output values for any given set of input parameters. The selection value is controlled by the switching signal ${}_k\sigma_i$, that is, the value ${}_kS_i \;\in\; \{1,\,0\}$ is one if ${}_k\sigma_i \;=\; n$, otherwise, the value is zero (cf. Fig. 1). The algorithm is periodically executed every control cycle, and it provides the next state of motion

$$ {}_k\vec{u}_{i+1}^{\,(n)} \;=\; ({}_kP_{i+1},\; {}_kV_{i+1},\; {}_kA_{i+1}) \quad (2) $$

that acts as command variable for the inner control loops.

Fig. 3 shows the decision tree of the algorithm, which is responsible for selecting the correct motion profile. Its development is rather straight-forward; the finite set of acceleration profiles consists of four elements (*PosTri*, *PosTrap*, *NegTri*, and *NegTrap*). The two intermediate profile segments *PosLin* and *NegLin* apply the positive or the negative maximum jerk value to guide the acceleration to $+{}_kA_i^{max}$, $-{}_kA_i^{max}$, or zero. The *PosTri* profile first applies $+{}_kJ_i^{max}$ without reaching $+{}_kA_i^{max}$ and then $-{}_kJ_i^{max}$ until $+{}_kA_N \;=\; 0$ is reached (i.e., the profile is *Triangle*-shaped). The *PosTrap* motion profile works in the same way, but here $+{}_kA_i^{max}$ is reached (i.e., *Trapezoid*-shaped). The *NegTri* and the *NegTrap* acceleration profiles work analogously with inverted signs [33].

B. Open-Loop Position/Pose Control

The input and output values of this algorithm are shown in Fig. 4. When used in Euclidian space, it is regarded as an open-loop pose controller otherwise as an open-loop position controller. In comparison to the input values of the velocity-based algorithm, a complete target state of motion for K DOFs at instant T_i is represented by the matrix

$$ \mathbf{U}_i^{trgt} \;=\; \left({}_1\vec{u}_i^{\,trgt},\; \ldots,\; {}_k\vec{u}_i^{\,trgt},\; \ldots,\; {}_K\vec{u}_i^{\,trgt}\right)^T \;, \quad (3) $$

that is, $\forall\; k \;\in\; \{1,\,\ldots,\,K\}$:

$$ {}_k\vec{u}_i^{\,trgt} \;=\; \left({}_kP_i^{trgt},\; {}_kV_i^{trgt},\; {}_kA_i^{trgt}\right) \;. \quad (4) $$

Furthermore, the kinematic motion constraints are defined as

$$ \mathbf{U}_i^{max} \;=\; \left({}_1\vec{u}_i^{\,max},\; \ldots,\; {}_k\vec{u}_i^{\,max},\; \ldots,\; {}_K\vec{u}_i^{\,max}\right)^T \quad (5) $$

with $\forall\; k \;\in\; \{1,\,\ldots,\,K\}$:

$$ {}_k\vec{u}_i^{\,max} \;=\; ({}_kV_i^{max},\; {}_kA_i^{max},\; {}_kJ_i^{max}) \;. \quad (6) $$

In the same way that the velocity-based algorithm of Fig. 2 calculated the state of motion for one DOF k, this second algorithm computes states of motion $\mathbf{U}_{i+1}^{(n-1)}$ for all selected DOFs. Whether a DOF is selected for open-loop pose/position control depends on the switching signal $\vec{\sigma}_i$, that is, if the k-th element ${}_k\sigma_i$ equals $(n-1)$, the algorithm of Fig. 4 generates output values for the DOF k (cf. Fig. 1).

The algorithm internally generates a trajectory to transfer the current state of motion $\mathbf{U}_i$ to the target state of motion $\mathbf{U}_i^{trgt}$ under consideration of constraints $\mathbf{U}_i^{max}$ in the shortest possible time (i.e., time-optimally). An important property is that all selected DOFs reach their target acceleration and their target velocity *synchronously* in their target position or pose at instant T_N (cf. eqn. (1)). As also mentioned in the previous subsection, it is of major importance that the algorithm is capable to compute output values for the entire input domain, that is, $\mathbb{R}^{9K} \times \mathbb{B}^K$ with $\mathbb{B} \;=\; \{0,\,1\}$.

IV. Stabilizing Hybrid Switched Motion Control Systems

This section introduces a little background information about discrete-time hybrid switched-system control and explains the method for stabilizing such systems.

A discrete-time hybrid switched-control system with K decoupled DOFs can generally be represented by a set of difference equations

$$ {}_k\vec{x}_{i+1} \;=\; {}_kf\left({}_k\vec{x}_i,\, {}_k\vec{u}_i^{\,({}_k\sigma_i)}\right) \;\forall\; k \;\in\; \{1,\,\ldots,\,K\}\;, \quad (7) $$

where ${}_k\vec{x}_i$ is the state of DOF k at instant T_i, ${}_k\vec{u}_i^{\,({}_k\sigma_i)}$ is the output of the control submodule with the index ${}_k\sigma_i$, and ${}_kf$ is a set of n distinct functions corresponding to the n control submodules of the system (cf. Fig. 1). ${}_k\sigma_i$ is a discrete signal that switches among the set $\{1,\,\ldots,\,n\} \subseteq \mathbb{N}$.

We have to ensure that

- if any closed-loop control submodule with an index $l \in \{1,\,\ldots,\,n-2\}$ is not able to control one or more DOFs of the system stably or
- if one or more DOFs of the system tend to be become instable due to inappropriate switching squences,

a *safe backup controller* can be provided that is capable to generate command variables for the lower-level controller from any arbitrary state. This is either done by

A) the velocity-based OTG algorithm (Sec. III-A) or
B) the position/pose-based OTG algorithm (Sec. III-B).

To apply this strategy, a set of possible criteria is required to detect an instability or a potential instability:

- The rate of switchings for one DOF k among the n control submodules is greater than a certain frequency threshold.
- The amplitude of one motion state variable (i.e., an element of ${}_k\vec{x}_i$) overshoots a predefined maximum value.
- The signal of a motion state variable transformed into the frequency domain is greater than a predefined maximum frequency.
- A sensor value (e.g., a measured force/torque value) is outside of the metering capacity of the sensor. In order to prevent any system or sensor damage, the system has to be switched to another safe control submodule.
- The motion system tends to leave its workspace.
- Maximum actuator forces and/or torques are reached.
- A sensor malfunction of one of the sensors in the feedback loop of one of the $(n-2)$ control submodules is detected.
- Further user-defined criteria and combinations of the above criteria (i.e., the user may define any undesired state to which he wants the system to react).

All these criteria can be freely set up by users. In the control cycle of instability detection T_d, in which one or more of these instability criteria become true for at least one DOF k, the switching signal ${}_k\sigma_d$ is either set to ${}_k\sigma_d = (n-1)$ or to ${}_k\sigma_d = n$. This depends on the stabilization strategy, such that one of the OTG algorithms *instantaneously* takes over control for the DOF k. As described in Sec. III, such an algorithm is able to generate a kinematically time-optimal trajectory within one control cycle, here: the control cycle, in which the instability was detected (i.e., between T_d and T_{d+1}).

Compared to the other $(n-2)$ control submodules, the OTG algorithms of the modules $(n-1)$ and n are the only ones that are capable to provide *steady command variables* for any motion state while taking motion constraints into account.

In any case, it is of major importance that the input parameters of the OTG algorithms (cf. Figs. 2 and 4) are adequate at the moment of switching. Using either solution, the discrete-time hybrid switched-system can be considered as a standard non-switched-system, such that the same methods as mentioned in Sec. II can be utilized during the design and parameterizing procedure of the overall system [1]–[6]. We can transform the problem of hybrid switched-system analysis to a stability analysis problem of a trajectory-following control scheme. The challenge and particularity of this analysis is that we have to consider an *arbitrary initial state of motion*: The state of motion that has been achieved by the control submodules at the instant the instability or potential instability is detected. This motion state depends on sensor signals. Thus, the analysis result only depends on one parameter (i.e., the motion state), that is, for a formal proof of stability, one would have to bound the allowed motion state, that is, the velocity and the acceleration values of the plant. These actuator space velocity and acceleration constraints are the same as the ones that have been used for the proof of Lyapunov stability of the actuator space control scheme. This line of arguments leads us to the (expected and trivial) result that we have to ensure that the resulting velocities and accelerations that are computed in or for the actuator space are limited during the entire operation time of the plant.

To emphasize it again: This is not a formal proof of stability but only a logical line of arguments that leads to the heuristic and plausible result of stability. Of course, this result is solely due to the OTG algorithm: *Before* the system becomes instable, one of the two backup modules, n or $(n-1)$, takes over control and generates command variables for the system being in any arbitrary state of motion.

V. Real-World Experimental Results

To highlight the practical relevance of the method proposed in the previous section, we now show real-world experimental results. For the experiments, the following hardware setup has been used: The original controller of a six-joint Stäubli RX60 industrial manipulator [35] was replaced, and the frequency inverters were directly interfaced. Three PCs running with QNX [36] as real-time operating system perform a control rate of 10 KHz for the joint controllers; a hybrid switched-system controller is used for Cartesian space control and runs at a frequency of 1 KHz. The output signals of the switching system

$$ \mathbf{U}_i \;=\; \left({}_x\vec{u}_i,\, {}_y\vec{u}_i,\, {}_z\vec{u}_i,\, {}_{\textcircled{x}}\vec{u}_i,\, {}_{\textcircled{y}}\vec{u}_i,\, {}_{\textcircled{z}}\vec{u}_i,\, \right)^T \quad (8) $$

are transformed into joint space, whereas the translational DOFs are indexed by x, y, z, the rotational ones by $\textcircled{x}$, $\textcircled{y}$, $\textcircled{z}$. The following two subsections explain the usage of the velocity-based OTG algorithm and of the pose-based OTG algorithm for this setup.

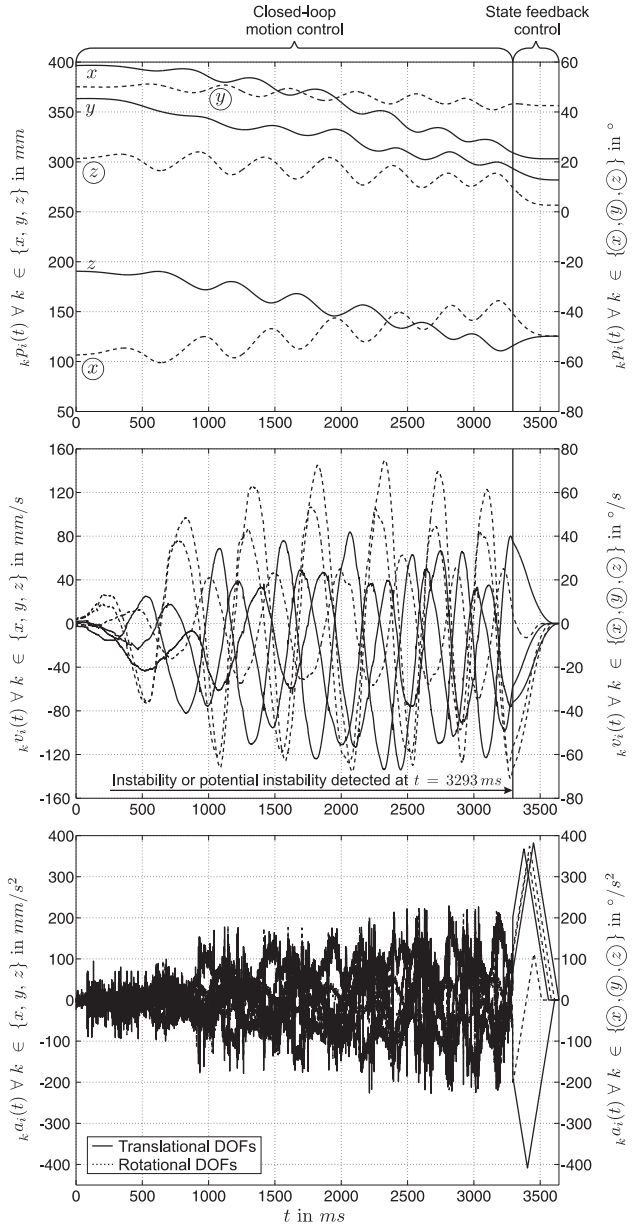

Fig. 5. Stabilizing a discrete-time hybrid switched-system with the velocity-based OTG algorithm of Sec. III-A: At $t = 3293\,ms$, the instability is detected (caused by the velocity signals), the system sets ${}_k\sigma_{3293} = n-1$ for all six DOFs, and the velocity-based OTG algorithm takes over control to guide all DOFs to zero-velocity. Fig. 6 shows the same trajectories for $3293\,ms \leq t \leq 3641\,ms$ in state space.

A. Stabilizing a Robot Manipulator with the Velocity-Based OTG Algorithm

Figs. 5 and 6 show, how the velocity-based OTG algorithm of Sec. III-A is used to stabilize an unstable system. As described in the previous section, it is of major importance to setup correct input parameters

$$\begin{aligned} &{}_kV_i^{trgt},\ {}_kA_i^{max},\ {}_kJ_i^{max} \\ &\qquad \forall\ (k,\,i)\ \in\ \left\{x,\,y,\,z,\textcircled{x},\textcircled{y},\textcircled{z}\right\}\ \times\ \mathbb{Z}\ . \end{aligned} \tag{9}$$

In the simplest case,

$$\vec{V}_i^{\,trgt} = \vec{0} \quad \forall\ i\ \in\ \mathbb{Z} \tag{10}$$

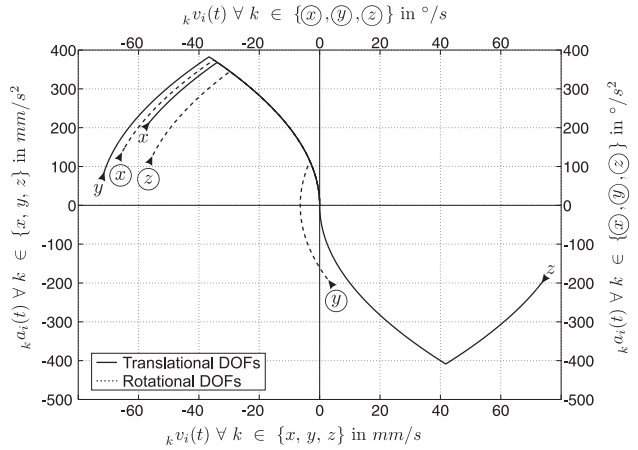

Fig. 6. On-line generated trajectories drawn in the velocity-acceleration-plane of the state space generated by the velocity-based OTG algorithm for the time interval $3293\,ms \leq t \leq 3641\,ms$ (corresponding to Fig. 5). The requirement for achieving stability here is to let all trajectories terminate in an equilibrium point of the underlying control loops.

holds. If we denote the maximum actuator forces or torques at an instant T_i as $\vec{F}_i^{\,max}$, we can calculate the maximum acceleration value from the differential equation of the forward dynamics [37]

$$\vec{A}_i^{\,max} = \vec{f}\left(\vec{P}_i,\,\vec{V}_i,\vec{F}^{\,max}\right)\ . \tag{11}$$

$\vec{J}_i^{\,max}$ may be determined in dependence on the current task. For the experiment of Figs. 5 and 6, first, a sensor-guided motion is executed, that is,

$$\begin{aligned} \forall\ (k,\,i)\ \in\ &\{\,x,\,y,\,z,\textcircled{x},\textcircled{y},\textcircled{z}\,\}\ \times\ \{0,\,\ldots,\,3292\}: \\ &{}_k\sigma_i\ \in\ \{1,\,\ldots,\,n-2\}\ . \end{aligned} \tag{12}$$

The potential instability of one of the control submodules is detected in the control cycle of instant T_{3293} due to the velocity signals in or more DOFs. At this instant, the OTG algorithm of Fig. 2 took over control to guide all DOFs to zero-velocity. The pose, velocity, and acceleration progression can be seen in Fig. 5; the respective trajectories in the velocity-acceleration-plane of the state space are shown in Fig. 6. As one can see here, the OTG algorithm itself can be regarded as a switching system that switches the jerk in-between

$$\begin{aligned} &\{-{}_kJ_i^{max},\,0,\,+{}_kJ_i^{max}\} \\ &\qquad \forall\ (k,\,i)\ \in\ \{\,x,\,y,\,z,\textcircled{x},\textcircled{y},\textcircled{z}\,\}\ \times\ \mathbb{Z}\ . \end{aligned} \tag{13}$$

To guarantee system stability, it is required that the Lyapunov stable inner control loops are guided to an equilibrium point. In this example, all six DOFs are switched simultaneously to the velocity-based OTG algorithm ($\vec{\sigma}_{3293} = n-1$); depending on the task, it would, of course, also be possible to switch only a selection of DOFs.

B. Stabilizing a Robot Manipulator with the Pose-Based OTG Algorithm

Here, we repeat the experiment, but instead of the velocity-based OTG algorithm, the pose-based one is applied; the

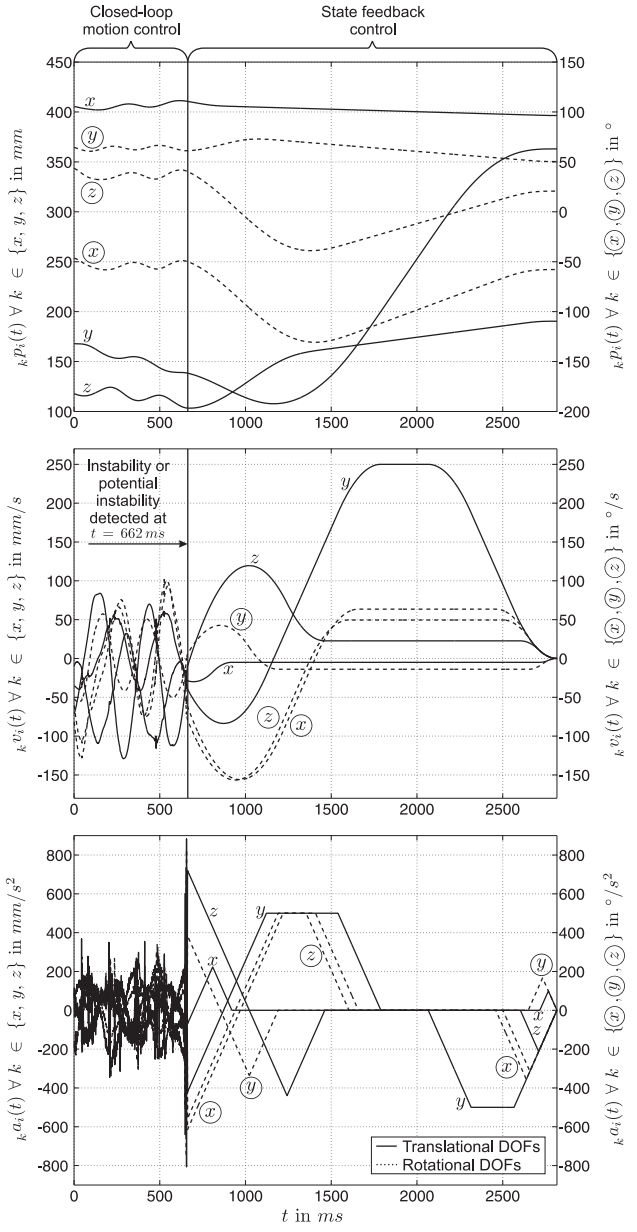

Fig. 7. Position, velocity, and acceleration progressions. The instability of the system is detected at $t = 662\,ms$, that is, the system switched from sensor-guided motion control to open-loop pose control (i.e., $\vec{\sigma}_{662} = (n,\, n,\, n,\, n,\, n,\, n)^T$). All six trajectories coinstantaneously reach their desired target state of motion at $t = 2816\,ms$. Fig. 8 depicts these trajectories in the velocity-acceleration-plane of the state space.

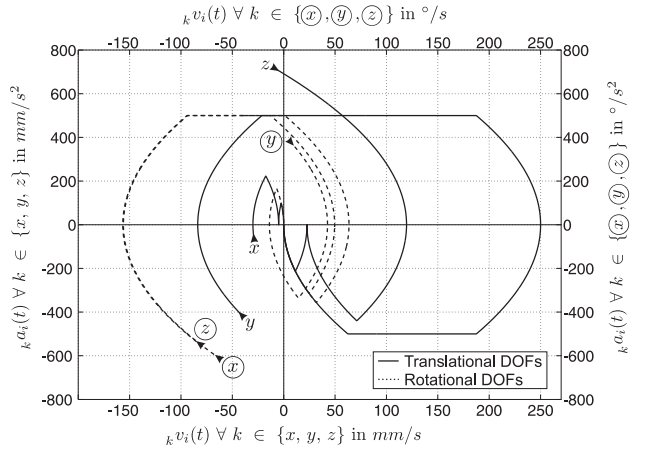

Fig. 8. Corresponding to Fig. 7, this diagram illustrates the six trajectories from the moment of switching on (i.e., in the interval $662\,ms \leq t \leq 2816\,ms$). As in Fig. 6, all trajectories terminate in an equilibrium point of the underlying control loops.

results are shown in Figs. 7 and 8. Starting with a sensor-guided motion, the instability is detected at $T_d = 662\,ms$ and was caused by the great amplitudes of the acceleration. As mentioned in the previous subsection, it is essential that appropriate input parameters $\mathbf{U}_i^{trgt}$ and $\mathbf{U}_i^{trgt}$ are setup in the moment of switching all elements of $\vec{\sigma}_i$ to n. $\vec{A}_i^{max}$ may be calculated from the local forward dynamics again (cf. eqn. (11) and [37]), $\vec{V}_i^{max}$ commonly is given through mechanical system properties, and $\vec{J}_i^{max}$ may be setup with regard to the current task again. The simplest way to setup $\mathbf{U}_i^{trgt}$ would be to choose

$$\vec{V}_i^{trgt} = \vec{0} \quad \text{and} \quad \vec{A}_i^{trgt} = \vec{0} \quad \forall\, i \in \mathbb{Z} \tag{14}$$

again. The desired pose $\vec{P}_i^{trgt}$ should be set to a safe pose in workspace, such that no collisions and no singularities occur during the motion. Depending on the task, it can also be reasonable to specify desired target velocity vectors in space $\vec{V}_i^{trgt}$ unequal to zero (e.g., in order to synchronize the system with cooperating one or to achieve a defined state of motion from which a safe motion can be continued).

Fig. 7 depicts the position, velocity, and acceleration progressions for all six DOFs $\{\, x,\, y,\, z, \textcircled{x}, \textcircled{y}, \textcircled{z} \,\}$, and Fig. 8 displays the corresponding trajectories the velocity-acceleration-plane of the state space from the moment of switching on. As one can clearly see in Fig. 7, the trajectories of all six DOFs are continuous, and they reach their desired target state of motion $\mathbf{U}_i^{trgt}$ coinstantaneously at $T_N = 2816\,ms$. Furthermore, Fig. 8 shows that all six trajectories terminate in an equilibrium point of the inner control loops as eqn. (14) was applied in this experiment (i.e., it is a Type III trajectory, cf. [33], [34]). As can be seen in Fig. 7, eqn. (13) holds again. Furthermore, it would also be possible to switch only a selection of DOFs instead of all DOFs.

VI. Conclusion

A method for stabilizing discrete-time hybrid switched motion control systems in multiple DOFs is presented. Two different on-line trajectory generation algorithms are introduced that are both able to generate trajectories from arbitrary states of motion and to time-optimally transfer the system to a desired state. The method consists of two steps:

1) Detecting an instability or potential instability.
2) Switching to an on-line trajectory generation algorithm and guiding the system to an equilibrium point.

Real-world results achieved with a six-joint industrial manipulator underline the practical relevance as safe switchings from sensor-guided motions (e.g., force/torque control, visual

servo control, or distance control) to trajectory-following motions that are possible now, such that it is *always* possible to reach a safe and stable state of motion. This is in particular relevant for robot manufacturers, because it is a safe strategy to react to sensor failures or malfunctions.

ACKNOWLEDGEMENTS

The authors would like to thank *QNX Software Systems* [36] for providing free software licenses.

REFERENCES

[1] W. Chung, L.-C. Fu, and S.-H. Hsu. Motion control. In B. Siciliano and O. Khatib, editors, *Springer Handbook of Robotics*, chapter 6, pages 133–159. Springer, Berlin, Heidelberg, Germany, first edition, 2008.

[2] K. Kozłowski. *Modelling and Identification in Robotics*. Springer, London, UK, 1998.

[3] L. Sciavicco and B. Siciliano. *Modelling and Control of Robot Manipulators*. Advanced Textbooks in Control and Signal Processing. Springer, London, UK, second edition, 2000.

[4] M. W. Spong, S. A. Hutchinson, and M. Vidyasagar. *Robot Modeling and Control*. John Wiley and Sons, 2006.

[5] C. C. De Wit, B. Siciliano, and G. Bastin. *Theory of Robot Control*. Springer, Berlin, Germany, 1996.

[6] W. Leonhard. *Control of Electrical Drives*. Springer, Berlin, Germany, third edition, 2001.

[7] S. Sastry. *Nonlinear Systems: Analysis, Stability, and Control*. Springer, New York, NY, USA, 1999.

[8] M. S. Branicky. *Studies in Hybrid Systems: Modeling, Analysis, and Control*. PhD thesis, Electrical Engineering and Computer Science Dept., Massachusetts Institute of Technology, `http://dora.cwru.edu/msb/pubs.html` (accessed: Dec. 15, 2008), 1995.

[9] M. S. Branicky. Multiple Lyapunov functions and other analysis tools for switched and hybrid systems. *IEEE Trans. on Automatic Control*, 43(4):475–482, April 1998.

[10] D. Liberzon. *Switching in Systems and Control*. Systems and Control: Foundations and Applications. Birkhäuser, Boston, MA, USA, 2003.

[11] D. Liberzon and A. S. Morse. Basic problems in stability and design of switched systems. *IEEE Control Systems Magazine*, 19(5):59–70, October 1999.

[12] R. W. Brockett. Hybrid models for motion control systems. In H. L. Trentelman and J. C. Willems, editors, *Essays on Control: Perspectives in the Ttheory and its Applications*, chapter 2, pages 29–53. Birkhäuser, Boston, MA, USA, first edition, March 1993.

[13] M. Žefran and J. W. Burdick. Stabilization of systems with changing dynamics by means of switching. In *Proc. of the IEEE International Conference on Robotics and Automation*, volume 2, pages 1090–1095, Leuven, Belgium, May 1998.

[14] M. Žefran and J. W. Burdick. Design of switching controllers for systems with changing dynamics. In *Proc. of the IEEE Conference on Decision and Control*, volume 2, pages 2113–2118, Tampa, FL, USA, December 1998.

[15] M. S. Branicky. Stability of hybrid systems: State of the art. In *Proc. of the IEEE Conference on Decision and Control*, volume 1, pages 120–125, San Diego, CA, USA, December 1999.

[16] M. A. Wicks, P. Peleties, and R. A. DeCarlo. Construction of piecewise Lyapunov functions for stabilizing switched systems. In *Proc. of the IEEE Conference on Decision and Control*, volume 4, pages 3492–3497, Lake Buena Vista, FL, USA, December 1994.

[17] P. J. Antsaklis, W. Kohn, M. Lemmon, A. Nerode, and S. Sastry, editors. *Hybrid Systems V*, volume 1567 of *Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg, Germany, New York, NY, USA, 1999.

[18] W. P. Dayawansa and C. F. Martin. A converse Lyapunov theorem for a class of dynamical systems which undergo switching. *IEEE Trans. on Automatic Control*, 44(4):751–760, April 1999.

[19] G. Michaletzky and L. Gerencsér. BIBO stability of linear switching systems. *IEEE Trans. on Automatic Control*, 47(11):1895–1898, November 2002.

[20] R. A. DeCarlo, M. S. Branicky, S. Pettersson, and B. Lennartson. Perspectives and results on the stability and stabilizability of hybrid systems. *Proc. of the IEEE*, 88(7):1069–1082, July 2000.

[21] N. H. McClamroch and I. Kolmanobsky. Performance benefits of hybrid control design for linear and nonlinear systems. *Proc. of the IEEE*, 88(7):1083–1096, July 2000.

[22] X. Xu and G. Zhai. Practical stability and stabilization of hybrid and switched systems. *IEEE Trans. on Automatic Control*, 50(11):1897–1903, November 2005.

[23] B. Finkemeyer, T. Kröger, and F. M. Wahl. Executing assembly tasks specified by manipulation primitive nets. *Advanced Robotics*, 19(5):591–611, June 2005.

[24] L. Villani, , and J. De Schutter. Force control. In B. Siciliano and O. Khatib, editors, *Springer Handbook of Robotics*, chapter 7, pages 161–185. Springer, Berlin, Heidelberg, Germany, first edition, 2008.

[25] N. Hogan. Impedance control: An approach to manipulation. Part I: Theory. Part II: Implementation. Part III: Applications. *ASME Journal of Dynamic Systems, Measurment, and Control*, 107:1–24, March 1985.

[26] S. Chiaverini and L. Sciavicco. The parallel approach to force/position control of robotic manipulators. *IEEE Trans. on Robotics and Automation*, 9(4):361–373, August 1993.

[27] M. H. Raibert and J. J. Craig. Hybrid position/force control of manipulators. *ASME Journal of Dynamic Systems, Measurement and Control*, 102:126–133, June 1981.

[28] F. Chaumentte and S. A. Hutchinson. Visual servoing and visual tracking. In B. Siciliano and O. Khatib, editors, *Springer Handbook of Robotics*, chapter 24, pages 563–583. Springer, Berlin, Heidelberg, Germany, first edition, 2008.

[29] N. R. Gans and S. A. Hutchinson. A switching approach to visual servo control. In *Proc. of the IEEE International Symposium on Intelligent Control*, pages 770–776, Vancouver, Canada, October 2002.

[30] N. R. Gans. *Hybrid Switched System Visual Servo Control*. PhD thesis, Department of General Engineering, University of Illinois at Urbana-Champaign, 2005.

[31] N. R. Gans and S. A. Hutchinson. Stable visual servoing through hybrid switched-system control. *IEEE Trans. on Robotics*, 23(3):530–540, June 2007.

[32] K. Kühnlenz and M. Buss. Stability issues in sensor switching visual servoing. In *In Proc. of SPIE, the International Society for Optical Engineering*, pages 671902.1–671902.14, Lausanne, Switzerland, October 2007.

[33] T. Kröger. *On-Line Trajectory Generation in Robotic Systems*, volume 58 of *Springer Tracts in Advanced Robotics*. Springer, Berlin, Heidelberg, Germany, first edition, 2009.

[34] T. Kröger and F. M. Wahl. On-line trajectory generation: Basic concepts for instantaneous reactions to unforeseen events. *Accepted for publishing in: IEEE Trans. on Robotics*, 2009.

[35] Stäubli Faverges SCA, Place Robert Stäubli BP 70, 74210 Faverges (Annecy), France. Homepage. `http://www.staubli.com/en/robotics` (accessed: Dec. 15, 2008). Internet, 2008.

[36] QNX Software Systems, 175 Terence Matthews Crescent, Ottawa, Ontario, Canada, K2M 1W8. Homepage. `http://www.qnx.com` (accessed: Oct. 23, 2009). Internet, 2009.

[37] R. Featherstone. *Rigid Body Dynamics Algorithms*. Springer, New York, NY, USA, first edition, 2007.