Hybrid System for Event-Based Planning and Control of Robot Operation

Karel Jezernik, Aleš Hace
University of Maribor, Faculty of Electrical Engineering and Computer Science, Maribor, Slovenia
karel.jezernik@uni-mb.si, ales.hace@uni-mb.si

Abstract - **The hybrid systems contain two distinct types of components, subsystem with continuous dynamics and subsystem with discrete dynamics that interact with each other. Such hybrid systems arise in varied contexts in manufacturing, communication networks, auto-pilot design, and traffic control and in robotics and mechatronics, among others. Hybrid systems have a central role in embedded control systems that interact with the physical world. They also arise from the hierarchical organization of complex systems and from the interaction of discrete planning algorithms and continuous control algorithms in autonomous, intelligent systems. The experimental results demonstrate the efficiency of the hybrid approach for the bilateral control of a 1-DOF master-slave teleoperation.**

I. Introduction

Hybrid systems, covering heterogeneous continuous and discrete event nature of the dynamical systems are well known academic approach. They are characterized by the interaction of continuous part governed by the differential or difference equations, and by discrete event part, traditionally described by finite state machines, if-then-else rules, temporal logic and discrete components, on-off switches, digital circuitry, software code, etc [1]. Hybrid systems switch among many operating modes, where each mode is governed by its own characteristic dynamical laws. Mode transitions are triggered by variable crossing specific thresholds – state events, by elapse of certain time periods – time events, or by external inputs – input events [2]. Typical hybrid systems are embedded systems, composed of dynamical components governed by logical/discrete decision components. Hybrid systems require interdisciplinary approaches exploiting formal methods in computer science, control theory and operation research. The development of a theory and general tools for controller synthesis, verification, stability, and performance analysis are still in the infancy [3]. Traditional control engineering practice privileges the recursive methods for description of control structures. Recursive structures are much simpler to design, analyze, evaluate and implement on processor and FPGA based control platforms. State space description of continuous and discrete time processes is a typical representative of such approach. A promising recursive description for event-driven systems has been introduced recently. This paper applies this recently developed recursive description of event-driven systems in the field of mechatronic systems [4].

The tasks of robotic manipulating systems usually consist of multiple segments of robotic actions, which involve both continuous and discrete types of actions. A robotic process usually consists of a set of control actions (tasks) of machine modules, such as moving a robot, pinching/planning a part, delivering parts, etc. In order to complete the desired process, all the controlled actions should be executed according to their logical and temporal relationship [5]. Hence, designing the automated robotic manufacturing system usually involves solving a three-layer problem, shown in Fig. 1:

- determining the ordered sequence of controlled actions, called task scheduling layer,
- planning the actions themselves such as path, trajectory, etc., called action planning layer,
- designing a control mechanism to ensure the execution of the action plans and to achieve the goals of the synchronized actions, called motion control layer.

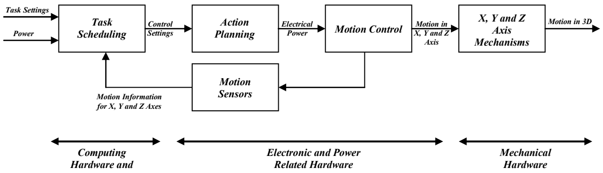

Fig. 1. Hybrid control architecture.

Traditionally, the task scheduling, action planning, and control are treated as separate issues. At the task-scheduling level, only discrete events are considered. The result of tasks scheduling is a sequence of logical commands. The focus of this paper is to present a new hybrid approach to the planning and control of bilateral operations [6].

II. Hybrid Systems Focus

A. Hybrid Control for Improved Closed-Loop Properties

The most common class of control architectures has, historically, been based on linear feedback. In the case of classical nonlinear controllers, there has been substantial growth in recent years in control design methods that are based on feedback linearization, passivity, adaptation, and control Lyapunov functions [7].

Hybrid controllers can be used to obtain improved closed-loop performance, beyond what can be achieved by using either classical linear or smooth nonlinear controllers. Motivation for hybrid control is the following: If the hybrid controller is appropriately defined, than the hybrid closed loop can reflect, to some degree, multiple performance properties associated with the closed-loop properties provided by each individual feedback function. This statement can be expressed in a slightly different way: motivation for use of hybrid control is that the performance of a hybrid closed loop can exceed the performance that can be achieved by any fixed feedback controller without switching. The results that we subsequently present illustrate the realization of this potential.

In this paper, hybrid controllers are represented by the block diagram in Fig. 2. That is, the hybrid control

978-1-4244-6886-7/10/$26.00 ©2010 IEEE

architecture consists of a family of nonlinear feedback functions and a supervisor; at each instant, the supervisor selects a particular feedback function from the family. The supervisor controls the feedback function selection as well as when the selection changes. Such hybrid controllers have often been called logic-based switching controllers and are the most widely studied class of hybrid controllers [8].

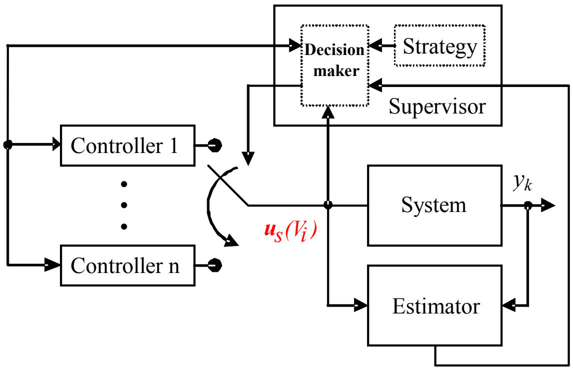

Fig. 2. Hybrid control architecture.

The supervisor selects a feedback function to be active by specification of control Lyapunov function [9]. The supervisor, as well as the family of feedback functions, may include dynamics and delays or other memory elements. The supervisor can be time driven, in which case switching is scheduled according to a clock as in a digital controller; or it can be event driven, in which case switching occurs according to a state partition or a state-dependent switching condition. This structure, for the supervisor, allows transitions between feedback function selections according to an automata model and common hybrid control architecture. It is important that the supervisor does not switch between feedbacks functions infinitely, often in a finite time period; this can be achieved in a number of ways. Design of the complete hybrid controller involves specification of both the family of feedback functions and specification of the switching logic. There is little guidance in the published literature on how the family of feedback functions should be selected; this is often problem dependent.

B. *Modelling of Disrete-Event Process*

Considering discrete event systems in computer science, a state space is a description of the structure of discrete states that the system can occupy. Switching among the discrete states is triggered by occurrence of events in the system. The structure of discrete states and events can be represented as a simple model of automata or state machine. Formally, it can be defined as a four-tuple [N, A, S, G] where:

- N is a set of states
- A is a set of arcs connecting the states
- S is a nonempty subset of N that contains start states
- G is a nonempty subset of N that contains the goal states.

An interesting matrix description of discrete event-driven processes is introduced recently in [10],[11]. The operation of discrete event system is presented by their discrete states ***m*** and discrete events ***x***. The system can take only one discrete state at the instant and the transition to other states are achieved by the occurrence of events, which represents a fulfillment of system conditions. Besides the states m and events x, the system operation is governed by inputs u and outputs y. Inputs u denote external conditions for occurrence of events x, while outputs y present output signals according to the current discrete state m of the system. The variables u, x, m and y, are Boolean vectors. Their components are represented by 0 and 1, where 1 means that the appropriate input is set, event has occurred, state is active or output is set, while 0 means the opposite. The discrete dynamics of such a system is governed by transferring the system among its specific discrete states. This is denoted using combination of algebraic and logic equations. Introducing such formalism, traditional coding efforts are significantly reduced on one hand, and the control algorithm can be off-line verified on the other hand. The matrix based state space description of discrete event driven system is schematically depicted in Fig. 3.

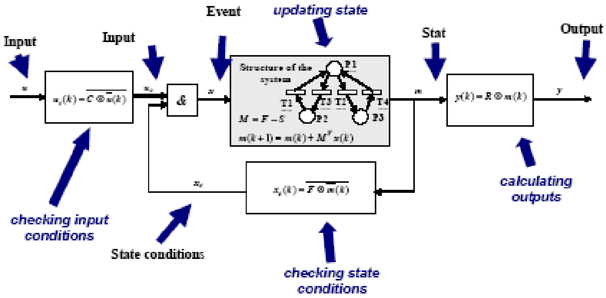

Fig. 3. Matrix based state space description of event-driven dynamics.

The recursive equations, representing the diagram in Fig. 3 are denoted by (1) - (4):

$$\boldsymbol{u}_c(k)=\overline{\boldsymbol{C}\otimes\overline{\boldsymbol{u}}(k)} \tag{1}$$

$$\boldsymbol{x}(k)=\overline{\boldsymbol{F}\otimes\overline{\boldsymbol{m}}(k)}\,\&\,\boldsymbol{u}_c \tag{2}$$

$$\boldsymbol{m}(k+1)=\boldsymbol{m}(k)+\boldsymbol{M}^T\boldsymbol{x}(k) \tag{3}$$

$$\boldsymbol{y}(k)=\boldsymbol{R}\otimes\boldsymbol{m}(k) \tag{4}$$

Equations (1), (2) and (4) are calculated using logical algebra. The over bar denotes logical negation. The symbol $\otimes$ denotes matrix Boolean product which is performed as a matrix multiplication, but logical “and” and “or” are applied instead of multiplication and addition respectively.

III. Motion Control

A. *Position Control of Servomechanism*

To design a controller, we consider a robot arm with second-order nonlinear differential equation

$$\boldsymbol{D}(\boldsymbol{q})\ddot{\boldsymbol{q}}+\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})+\boldsymbol{G}(\boldsymbol{q})=\boldsymbol{T} \tag{5}$$

where $\boldsymbol{q}$, $\boldsymbol{D}(\boldsymbol{q})$, $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})$, $\boldsymbol{G}(\boldsymbol{q})$, and $\boldsymbol{T}$ are, respectively, the joint angle vector, the inertia matrix, the term of centripetal and Coriolis forces, the gravity, and the joint torque. The joint torque $\boldsymbol{T}$ must satisfy the constraint

$$T_i^{\min}(\boldsymbol{q},\dot{\boldsymbol{q}}) \le T_i \le T_i^{\max}(\boldsymbol{q},\dot{\boldsymbol{q}}) \tag{6}$$

where i is the joint number. The output function in the robotic task space can be written as $\boldsymbol{Y} = \boldsymbol{H}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{x} & \boldsymbol{\Phi} \end{bmatrix}^T$, where $\boldsymbol{x}$ and $\boldsymbol{\Phi}$ are the robot's position and the orientation in the robot task space.

In order to obtain smooth mechanical motion of the robot the continuous desired trajectories with values $q^d, \dot{q}^d$ and $\ddot{q}^d$ are prescribed. The dynamic function, which determines the mechanical motion, is chosen to be of second order and a function of angular position, velocity and acceleration errors:

$$\sigma = \left(\ddot{q}^d - \ddot{q}\right) + K_v(\dot{q}^d - \dot{q}) + K_p(q^d - q), \tag{7}$$

where K_v and K_p are control design parameters that in the manifold mode ($\sigma = 0$) determine the prescribed dynamics of second order system.

B. Load Estimation

For the practical control implementation the measured quantities are state variables $q,\ \dot{q}$. The acceleration signal $\ddot{q}$ is not measurable and can be obtained by double differentiation of the angular position q, but it is contaminated by the measurement noise to such a degree that it can no longer be used. Consequently, the acceleration signal $\ddot{q}$ needs to be replaced by an estimated value $\hat{\ddot{q}}$, which is obtained simply from the differential equation of motion

$$\bar{J}(q)\hat{\ddot{q}} = T_e - \hat{T}_l, \tag{8}$$

where $\bar{J}$ is the mean value of J, T_e is the active measurable drive torque developed by the actuator and $\hat{T}_l$ is the estimate value of the load torque. The expression (8) is inserted into the control scheme (7) by replacing the real load torque T_l with an estimated value $\hat{T}_l$. An estimator of reduced order is:

$$\hat{T}_l = l\left(\dot{q}^c - \dot{q}\right), \tag{9}$$

where l is a positive constant linked to the selected dynamics of the asymptotic load observer and $\dot{q}^c$ is the calculated angular speed. The calculated angular acceleration signal $\ddot{q}^c$ is derived from (7), so the condition for the switching mode operation ($\sigma = 0$) of the system is fulfilled.

$$\ddot{q}^c = \ddot{q}^d + K_v(\dot{q}^d - \dot{q}) + K_p(q^d - q), \tag{10}$$

$$\dot{q}^c = \int_0^t \ddot{q}^c \, d\upsilon . \tag{11}$$

As a result the control input is based on a modified switching function which contains the estimated acceleration and the estimated disturbance torque

$$\begin{aligned} \sigma^e &= T_e^d - T_e = \\ &= \left[\ddot{q}^d + K_v\left(\dot{q}^d - \dot{q}\right) + K_p\left(q^d - q\right) + \hat{T}_l / \bar{J}\right]\hat{J} - T_e \end{aligned} \tag{12}$$

where the desired trajectory of angular position, velocity and acceleration is denoted by the superscript d and $\hat{T}_l$ is the estimated disturbance torque. The asymptotic observer serves as a bypass for high frequency components, therefore the unmodelled dynamics is not exceeded. Then estimated load torque (9) include

$$\hat{T}_l = \left(J - \bar{J}\right)\ddot{q} + \left(K_{tn} - K_t\right)I^d + C\left(q,\dot{q}\right) + G\left(q\right), \tag{13}$$

where K_t is a torque coefficient and I^d is motor current reference value. By feedback of the estimated load torque, robust motion control is accomplished and ideal acceleration control is achieved [12].

C. Bilateral Control

In bilateral control, high robustness to disturbance is required since bilateral systems are envisioned to be utilized in remote planes and are made control with unknown object during operation. This research utilized a bilateral controller proposed in [13]. Bilateral controller divides the system into two coordinates: common and differential. Position control with load estimator is applied to the differential coordinate, while force control with the use of reaction torque observer, shown on Fig. 4, is used in the common coordinate. Input of an operator and environmental force in simulation are presented in a spring and damper model. In this research we do not need to consider gravity torque influence since the experimental system is a 1-DOF rotary manipulator and set horizontally.

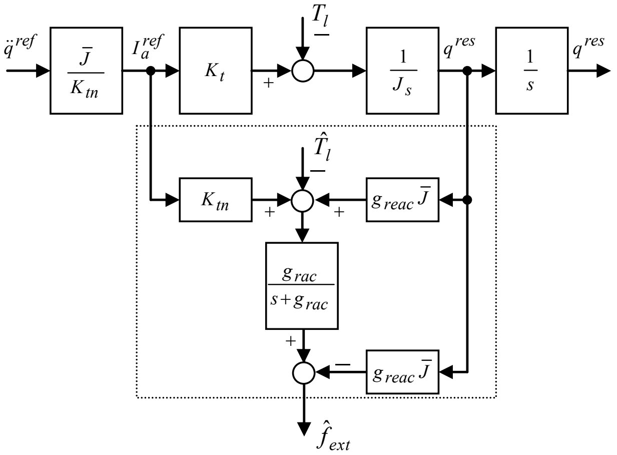

Fig. 4. Reaction torque estimation by RTOB.

IV. Event-Based Planning and Control

A. Task Scheduling

The event-based planning and control scheme is represented in Fig. 5. In Fig. 5, the function of the "Action Reference" block is to compute the scalar action reference variable based on the system output measurement. The planner then gives the desired input to the system based on the computed action reference. It can be seen that the planning becomes an investigation/decision component in the sense that it utilizes the output information and plans accordingly. Therefore, the event-based planning and control scheme has an ability to deal with unexpected or uncertain events. Based on the action reference variable s, each segment of the task can be planned as $q^d = q(s)$, $\dot{q}^d = \dot{q}(s)$, and $\ddot{q}^d = \ddot{q}(s)$. It describes a desired motion or action plan as a function of the action reference variable [14].

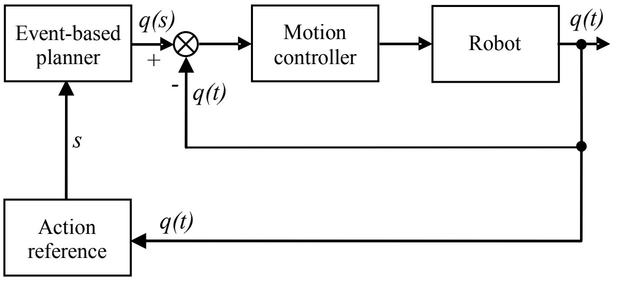

Fig. 5. Event-based planning and control.

In addition, the action reference variable is calculated at the same rate as feedback control. In other words, the original plan is adjusted and modified at a very high rate. As a result, it is able to deal not only with discrete unexpected or uncertain events, but also with continuous unexpected and uncertain events, such as modeling error, system parameter drifting, etc,[15].

B. Discrete-Event Torque/Current Control

The considered control problem is the tracking of a three phase current reference signal [16]. After defining current control error $\Delta \boldsymbol{i}_s = \boldsymbol{i}_s - \boldsymbol{i}_s^d$, rewritten in error form becomes

$$L\frac{d}{dt}\Delta \boldsymbol{i}_s + R\,\Delta \boldsymbol{i}_s = \boldsymbol{e}_s - \boldsymbol{u}_s(V_i)\,, \qquad (14)$$

which collects all the disturbances (exogenous and endogenous) action on the system.

The basic principle of the current control is to manipulate the output voltage vectors so that the desired current of three phase inverter circuits is produced. This is achieved by choosing a converter switch combination $\boldsymbol{S}_j$ that drives the motor current vector by directly applying the appropriate voltages $\boldsymbol{u}_s(V_i)$. The switch positions of the three-phase converter are described using the logical variables $\boldsymbol{V}_\mathrm{i}$, dependent if switch $\boldsymbol{S}_\mathrm{i}$ is ON or OFF. Each variable corresponds to one phase of the converter. Three-phase converter can produce 2^3 voltage vector combinations; two of them are zero vectors and 6 active vectors, Fig. 6, [17].

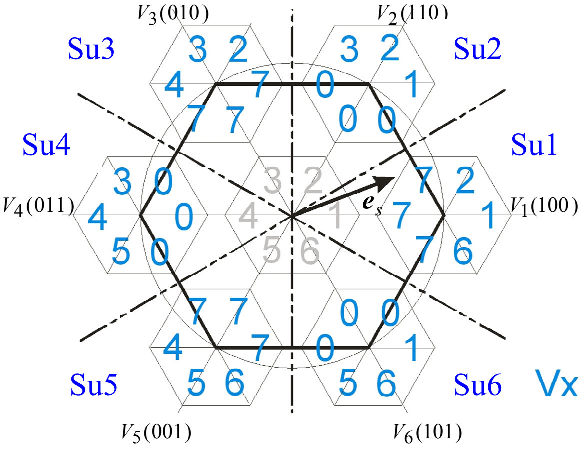

Fig. 6. Output voltage vector $\boldsymbol{u}_s$ sector allocation spaces..

The energy flow between the input and output side of the three phase converter is controlled by switching matrix. By introducing the binary variables S_i that are "1" if particular switch S_i is ON and "0" if switch S_i is OFF (i=1,2,3,…,6) the behavior of the switching matrix can be described by the three dimensional vector $\boldsymbol{u}_s = U_{DC}\,\boldsymbol{L}\,\boldsymbol{S}_i$, where matrix $\boldsymbol{L}$ and vector $\boldsymbol{S}(S_1, S_2, S_3)$ are defined as [18]

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \end{bmatrix} . \qquad (15)$$

$$\mathbf{S}^T = \begin{bmatrix} S_1 & S_2 & S_3 & \bar{S}_1 & \bar{S}_2 & \bar{S}_3 \end{bmatrix}$$

Relation (15) is true for the switching matrix $\boldsymbol{L}$. It essentially shows that this particular switching matrix is able to generate three independent control actions denoted as the components S_1, S_2 and S_3 of the control vector $\boldsymbol{u}_s(V_i) = U_{DC}\left[S_1, S_2, S_3\right]^T$. The components, i.e. switch position of the converter are generated by look up table of FPGA controller (Table I and Fig. 6).

TABLE I.
LOOK-UP TABLE

sign E / sign Di		Su1	Su2	Su3	Su4	Su5	Su6
		100	110	010	011	001	101
Sdi0	000	V7	V0	V7	V0	V7	V0
Sdi1	100	V1	V1	V7	V0	V7	V1
Sdi2	110	V2	V2	V2	V0	V7	V0
Sdi3	010	V7	V3	V3	V3	V7	V0
Sdi4	011	V7	V0	V4	V4	V4	V0
Sdi5	001	V7	V0	V7	V5	V5	V5
Sdi6	101	V6	V0	V7	V0	V6	V6
Sdi7	111	V7	V0	V7	V0	V7	V0

Considering a hysteresis controller as a discrete-event dynamical system, it allows forming in much more details on the switching actions and will enable a better understanding of the controller design. A discrete-event system reacts only if an event is recognized. To control the current $\boldsymbol{i}_s$, the sector of voltage $\boldsymbol{e}_s$ is recognized first, and based on the known sector, the output voltage vector $\boldsymbol{u}_s(V_i)$ (the transistor switching pattern) for the current control is selected respecting the current control error related to Lyapunov stability condition. Considering space vector representation of the induced voltage $\boldsymbol{e}_s$, the voltage is represented as vector rotates around the origin. Six active switching vectors of the three phase transistor converter result in six active output voltage vectors denoted $V_1 \ldots V_6$; V_0 and V_7 are two zero voltage vectors. According to signs of the phase voltages u_{s1} , u_{s2} and u_{s3} , the phase plane is divided into six sectors denoted Su1 … Su6, Fig. 6.

Regarding the situation, the voltage space vector $\boldsymbol{e}_s$ is in sector 1. In this sector, voltage vectors V_0 , V_1 , V_2 , V_6 and V_7 are selected for the current control. V_0 , V_7 are two zero vectors, while V_1 , V_2 , V_6 are three nearest adjacent live output voltage vectors to this sector. With the use of the discrete event system theory, five output voltage vectors $\boldsymbol{V}_0$, $\boldsymbol{V}_1, \boldsymbol{V}_2, \boldsymbol{V}_6$ and $\boldsymbol{V}_7$ are recognized as discrete states of the system. Events represent allowed transition among the discrete states i.e. allowed switching. The structure of the proposed strategy is represented by Petri Net graph on, Fig. 7. The Petri net formalism is propriety for research on possible deadlock or livelock of discrete event system under study. Switching among the available output voltage vectors in each sector is determined by the conditions that originate from the derivative of the Lyapunov function [19].

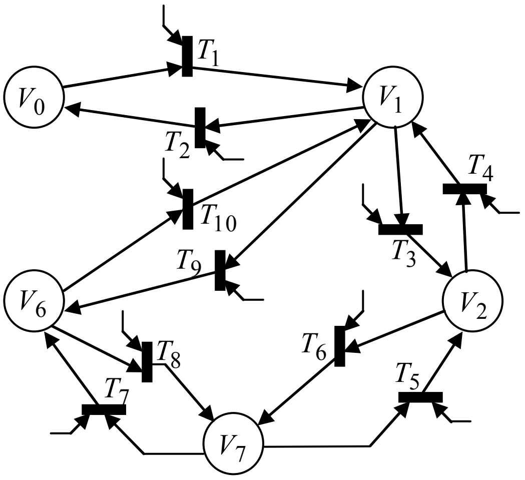

Fig. 7. PN-graph of the switching sequence in Sector 1.

For the Lyapunov function candidate

$$V = (1/2)\Delta \boldsymbol{i}_{\mathbf{s}}^{T} \Delta \boldsymbol{i}_{\mathbf{s}} = (1/2)\left(\boldsymbol{i}_s - \boldsymbol{i}_s^d\right)^T \left(\boldsymbol{i}_s - \boldsymbol{i}_s^d\right), \quad (16)$$

the stability requirement will be fulfilled if control can be selected as such, that the derivative of the Lyapunov function candidate is negative $\dot{V} = \Delta \boldsymbol{i}_s^{T} \Delta \dot{\boldsymbol{i}}_s \leq 0$. Derivatives of current control error (14) may be expressed with voltage equation

$$(d/dt)\left(\boldsymbol{i}_s - \boldsymbol{i}_s^d\right) = (1/L)\left(\boldsymbol{e}_s - R\,\boldsymbol{i}_s - \boldsymbol{u}_s(\boldsymbol{V}_j)\right) - (d/dt)\boldsymbol{i}_s^d, \quad (17)$$

where $\boldsymbol{i}_s^d$, $\boldsymbol{i}_s$ are desired and actual motor current, $\boldsymbol{u}_s(\boldsymbol{V}_j)$ is voltage control input, $R\,\boldsymbol{i}_s$ is resistive voltage drop and $\boldsymbol{e}_s$ is ac input voltage.

The conditions for the sequential switching of the power converter are selected as:

$$\begin{aligned} &S_1 = (1/2)\left(1-\operatorname{sign}(A)\right),\; S_2 = (1/2)\left(1-\operatorname{sign}(B)\right), \\ &S_3 = (1/2)\left(1-\operatorname{sign}(C)\right) \end{aligned} \quad (18)$$

where

$$\begin{aligned} A &= \left(i_{sa} - i_{sa}^d\right) \\ B &= -(1/2)\left(i_{sa} - i_{sa}^d\right) - (\sqrt{3}/2)\left(i_{sb} - i_{sb}^d\right), \quad (19) \\ C &= -(1/2)\left(i_{sa} - i_{sa}^d\right) + (\sqrt{3}/2)\left(i_{sb} - i_{sb}^d\right) \end{aligned}$$

which is evolved from the Lyapunov function derivative. When U_{DC} has enough magnitude that $\dot{V} \leq 0$, than $V \to 0$ and $\boldsymbol{i}_s \to \boldsymbol{i}_s^d$. S_1, S_2 and S_3 represent the switching state of the three-phase power converter. In case when their value is 1, the upper transistor in the leg is turned on, whereas in the case the value is 0, the lower transistor is ON. Notice that if S_1, S_2, S_3 equal to zero simultaneously, no current is delivered to the motor.

The proposed logical event-driven motor current control can be realized in the form described in Table I, where states of current control error are presented by *sign(D_j)* ($D_j = S_1$, S_2, S_3) and currently active voltage sector is presented by *sign E* (e_{S1}, e_{S2}, e_{S3}). To further improve the presentation, active voltage vectors are marked in Table I with a blue background. Because the transition between converter switch states is performed by switching only one converter leg, converter switching frequency chattering are reduced.

V. The Experimental System

A. Experimental system

The efficiency of the proposed SM bilateral control scheme was demonstrated by the experimental system shown in Fig. 8. It consisted of two identical 1-DOF robot manipulators with a handle connected to the motor axis via a planetary gear head. The handles can either be used to operate the system by a human on master side or to provide the environment contact on the slave side. The handle rotational plane was in horizontal orientation that provides zero gravitational effects and thus friction present in the motors as well as the gear heads is the main disturbance effect. The master-slave manipulator system output positions that were measured by encoders mounted on the motor axes, while the velocity was obtained by traditional discrete differentiation and filtering of the position signals within the computer controller which processes the data and set the analog voltage reference value to the servo drivers. The reference value represents signal of desired current that the servo drivers were injected to the BLDC servomotors. The motor torque is considered to be proportional to the motor current.

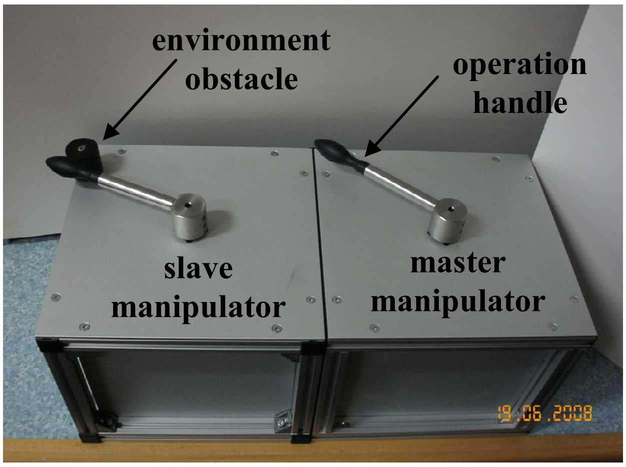

Fig. 8. Experimental bilateral system.

B. The bilateral controller setup

The master and slave controllers were implemented by the bilateral control and with $D_s = D_m = D$ and $M_m = M_s = 2M$; M was chosen so that matches total inertia in the manipulator. Although external force information was also necessary for the 4-channel bilateral architecture, force sensors were not used within this system. Instead, the external force observer was applied to estimate external force. By application of the external force observer, the estimated force signal can be described by

$$\hat{f}_{ext} = \frac{g}{s+g} f_{ext} \qquad (20)$$

where g denotes the cut-off frequency of the force observer. Table II shows the manipulators parameters and the control parameters are shown in Table III.

TABLE II.
MANIPULATOR PARAMETERS

Motor torque	*Nm*	0.355
Motor power	*W*	80
Motor rotor inertia	*gcm²*	20
Encoder resolution	*lines/rev*	500
Gearhead reduction ratio		14:1
Gearhead mass inertia	*gcm²*	0.8
Max. torque at gear output	*Nm*	3
Handle inertia	*kgcm²*	5.5
Handle length	*cm*	13.5

TABLE III.
BILATERAL CONTROL PARAMETERS

velocity filter cutoff freq.	*rad/s*	250
external force observer cutoff freq. g	*rad/s*	125
robust gain D	*1/s*	125
position gain k_p	*1/s²*	5000
velocity gain k_v	*1/s*	100
control rate	*Hz*	2000

C. Results

Experimental results are given by Fig. 9-Fig. 10. The top diagrams depict position response determined by handles angle trajectories of master and slave, respectively. The bottom diagrams depict force response indicated by the handles torque and estimated by the master and slave external torque observer, respectively.

Fig. 9 shows free motion experiment. In free motion only human operator holds the master handle, while the slave handle is not constrained by environment. Thus, in slow motion only low force at the master handle was required, whereas for high speed oscillating the human driving force was significantly increased to compensate for the inertial force within the experimental teleoperator. However, in both regimes almost perfect position tracking was observed.

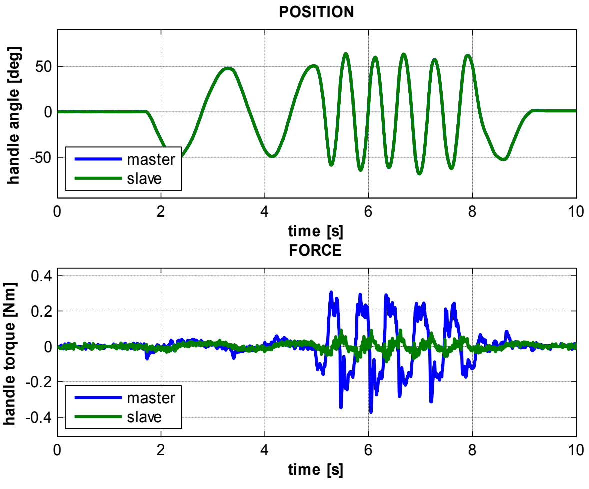

Fig. 9. Bilateral teleoperation experiment in free motion.

Fig.10 shows the experiment in which the slave handle was in contact with soft obstacle (sponge). Simultaneous position and force tracking was observed. More specific, force tracking was almost perfect, whereas position tracking was slightly deteriorated due to the fast decreasing of contact force.

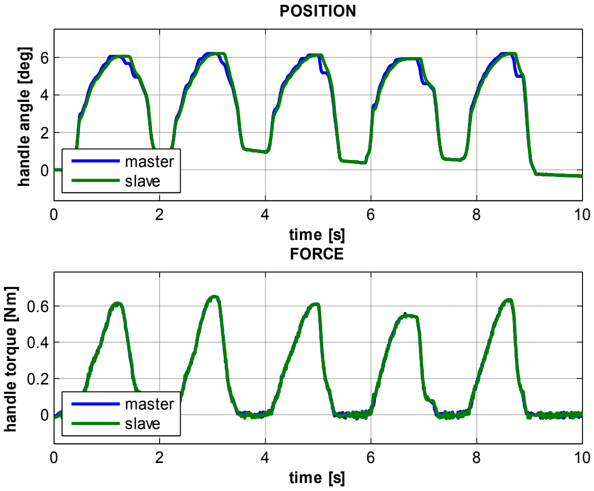

Fig. 10. Bilateral teleoperation experiment in soft contact motion.

VI. CONCLUSION

A new integrated task-scheduling, action-planning, and control method for a robotic manufacturing system has been proposed. Through a unified action reference variable, the event-based planning and control schemes has been extended to multisegment tasks. The use of the hybrid discrete event model along with the event-based planning and control provides a framework for the interaction of task scheduling, action planning, and control. This facilitates the analytical integration of low-level system sensing and control with high-level system behavior and preparation. As a result, the system is capable of coping with uncertainty and unexpected events and achieves higher efficiency and robust and reliable performance for automated robotic manufacturing systems.

The control scheme for bilateral robot operation is based on position and force (impedance) control approach, respectively, and is simple and easy to implement with the potential to provide high robustness against the disturbance, and thus it could be applied for high-accuracy haptic display in bilateral applications. It was experimentally validated for a 1-DOF master-slave teleoperators.

REFERENCES

[1] C.G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, 2nd ed. Springer, 2008.

[2] P. J. Ramadge, W. M. Wonham, "Supervisory Control of a Class of Discrete Event Process", *SIAM J. Control and Optimization*, vol. 25, no. 1, pp. 206-230, 1987.

[3] R. M. Jensen, "DES Controller Synthesis and Fault Tolerant Control. A Survey of Recent Advances", IT University Technical Reports Series, Copenhagen, 2003.

[4] E. G. Bryan, "Control Logic Requirements for Complex Manufacturing Systems," in *NSF Workshop on Logic Control for Manufacturing systems,* 2000.

[5] M.V. Iordache and P.J. Antsaklis, “Supervision Based on Place Invariants: A Survey”, ISIS Technical Report ISIS-2004-003.

[6] T.J. Tarn, AK. Bejcza, C. Guo, and N. Xi, “Intelligent Planning and Control for Telerobotic Operations”, *Proc. 21st International Conference on IEEE IECON*, vol.1, pp. XXXIV – XXXXV, 1995.

[7] G.A. Wainer: *Discrete-Event Modeling and Simulation: A Practitioner's Approach*. CRC Press, Taylor and Francis Group, Boca Raton, 2009.

[8] P.J.G Ramadge. and W.M Wonham, “The control of Discrete Event systems”, *Proceedings of the IEEE*, vol.77 no. 1, pp. 81-99, 1989.

[9] M.S. Branicky, V.S. Borkar and S.K. Mitter, “A unified framework for hybrid control: model and optimal control theory,” *IEEE Trans. Autom. Control.,* vol. 43, no. 1, pp. 31-45, Jan. 1998.

[10] A. Polič and K. Jezernik, “Closed-loop matrix based model of discrete event systems for machine logic control design”, *IEEE Trans. Industrial Informatics*, vol. 1, no. 1, pp. 39-46, feb. 2005.

[11] A. Polič, “*Matrix based approach to control of discrete event systems: a recursive description of logically controlled systems*”, PhD Thesis, University of Maribor, Maribor, Slovenia, 2005.

[12] K. Jezernik and M. Rodic, “High Precision Motion Control of Servo Drives”, *IEEE Tran. Ind. Electronics*, vol. 56, no. 10, pp. 3810-3816, 2009.

[13] K. Natori, T. Tsuji, K. Ohnishi, A. Hace, and K. Jezernik, “Time-Delay Compensation by Communication Disturbance Observer for Bilateral Teleoperation Under Time-Varying Delay “, *IEEE Tran. Ind. Electronics*, vol. 57, no. 3, pp. 1050-1062, 2010.

[14] J. Tan, N. Xi, and Y. Wang, “Integrated Task Planning and Control for Mobile Manipulators”, *Int. Journal of Robotics Research,* vol. 22, no. 5, pp. 337-354, May 2003.

[15] X. Li, C. Yang, Y. Chen, and X. HU, “Hybrid Event Based Control Arhitecture for Tele-robotic Systems Controlled through Internet”, *Journal of Zhejiang Univ. SCI,* vol. 5, no. 3, pp. 296-302, 2004.

[16] S.M. Eirea and G. Koo, “Hybrid modeling and control of power electronics”, *Proc. 6th Int. Workshop on hybrid systems: Computation and control,* vol. 2993 of Lecture notes in computer science, Springer, pp. 450-465, 2003.

[17] A. Polič, and K. Jezernik, “Event-driven current control structure for a three phase inverter”, *Int. rev. electrical eng. (Testo stamp.)*, vol. 2, no. 1, pp. 28-35, Jan.-Feb. 2007.

[18] A. Sabanovic, K. Jezernik and N. Sabanovic, “Sliding Modes Applications in Power Electronics and Electrical Drives”, In *Lecture notes in control and information sciences,* 1st ed., Y.U. Xinghuo, X.U. Jian-Xin, (Ed.). Berlin: Springer, pp. 223–251, 2002.

[19] E. Park, D.M. Tilbury, and P.P. Khargonekar, “A Formal Implementation of Logic Controllers for Machining Systems using Petri Nets and Sequental Function Chart”, *Proc. of the 1998 Japan- USA Symposium on Flexible Automation*, Otsu, Japan, 1998.