1918
TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY
FACULTY OF POWER ENGINEERING

9th International Symposium

TOPICAL PROBLEMS IN THE FIELD OF ELECTRICAL AND POWER ENGINEERING

◆

DOCTORAL SCHOOL OF ENERGY AND GEOTECHNOLOGY II

Pärnu, Estonia
June 14–19, 2010

9th International Symposium
„Topical Problems in the Field of Electrical and Power Engineering"
Pärnu, Estonia, June 14-19, 2010

Algorithm of Precise Control of Timetable for Intelligent Embedded Devices in City Electric Transport

Andrejs Potapovs, Mikhail Gorobetz, Anatoly Levchenkov, Leonīds Ribickis
Riga Technical University
Andrejs.Potapovs@rtu.lv mihails.gorobecs@rtu.lv anatolijs.levcenkovs@rtu.lv leonids.ribickis@rtu.lv

Abstract

Precise control of timetable is an important task in urban public transport industry.

Therefore it is useful to develop intelligent embedded control devices and control algorithms for city electric transport with high precision of performance of following the schedule.

This paper is addressed to the design process of direct control algorithm for embedded programmable device.

Keywords

electric transport; embedded systems; intelligent control

Introduction

In this paper is proposed algorithm of target braking for city electric transport. It allows significantly increasing of the electrical transport routing time and place accuracy, helping the driver to drive a tram directly to the driving speed, which guarantees the most accurate tracking of stop-time schedule by intelligent control system.

Development of the control algorithm is an important step in the overall system, because effectiveness of the program solution for controllers depends on its precision and accuracy.

Therefore action plan of algorithm is necessary with a high precision of prescription of the algorithm steps.

This factor is important because the exact algorithm in action planning, using textual description describes the sequence of steps and flow chart of algorithm shows the exact location of the step.

Automated and intelligent control systems provide large development opportunities in the future in electric transport systems and other complex electrical systems.

1. Purpose and Tasks

The purpose of the paper is design and realization of the accurate guidance system algorithm of electric transport.

The main tasks are:

- Definition of actions of proposed control system;
- Design and description of the control algorithm;
- Testing of algorithm realization for controllers on functional model and computer model of the electric transport system.

2. Mathematical Description of the control system

The city electric transport system consists of the following objects:

- T – set of electric vehicles (trams, trolleybuses)
- P – set of passenger stops – P_1, P_2, P_3

Each electric transport vehicle is equipped by the following elements:

- E - electric engine
- C – controller

One of the main hubs of the proposed control system is controller C that is installed in the tram, and processing all the information necessary to regulate the movement of the tram by control of tram electric engine E. This operation requires a controller program, after which it also will make all necessary calculations.

Following parameters of the electric transport vehicle are controlled:

- T – set of electric vehicles (trams, trolleybuses)
- V_T – tram speed, m/s
- S_t – tram carried path, m
- $t_{tekošais}$ – current time, s
- S_{TP} – the distance between tram and passengers stop, m
- S_{br} – breaking distance, m
- t_{bremz} – breaking time, s
- V_{nepiec} – required speed, m/s
- t_{atl} – the remaining travel time, s
- U_{imp} - voltage pulse width, %

Each passenger stop P has following set of parameters:

- X_P – coordinate of the passenger stop
- t_p – scheduled time of arrival

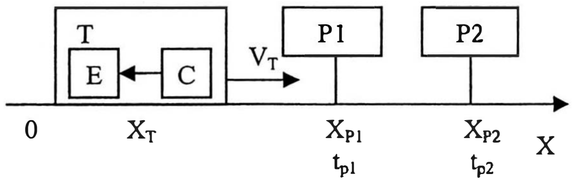


Fig. 1. General structure of the system

Fig. 1 presents general structure of the controlled systems with main control parameters.

Controller *C* of the electric vehicle T has two main blocks:

- *DB* – data base with a route schedule and position data of passenger stop
- *PG* – software for control

Tram has a particular route (fig. 1.), which will cross over a number of stops $P = (P_1, P_2, \ldots)$ and arrives in the exact time specified $t_p = (t_{p1}, t_{p2}, \ldots)$.

Stop coordinates X_p may be realized as a distance S_t from the route departure point $X = 0$.

This data are saved to controller's data base *DB*, which can provide a large enough number of stops on the data storage, allowing it to enter the whole movement stops on the route list. Each road section between points will be used directly to stop the upcoming data.

The administered control system is using needs additional equipment of the tram:

- positioning sensor
- speed sensor

Positioning sensor gets position of the electric transport.

Tram control program using the data to meet the approaching stop activities that one way or another be related to the movement of the tram ride in the required speed or stopping exactly stop.

The required speed of calculation principle will be as follows. Knowing the time at which the tram is due to arrive on the halt, and the rest of the way to passenger stop this will be calculated for average speed, which should take the remaining road section. In case of unexpected stops (stopper, the barrier material or compounding delays in the stop, etc) needed to be converted into speed and movement will take place with this new highest speed. This required the speed calculation will keep the program at each processing cycle, which will provide fast and accurate speed needed for a change.

Programs for the purpose of stopping in the test will simulate a supply of specially intended for the controller input signal, which will drop the engine speed, as can be observed after the speed increase, compared to a rate that was before the unexpected braking.

The stopping principle will be that, if fulfilled the condition that the distance to the passenger stop is less than the distance by tram need to go apply on the brake it to a certain braking acceleration (which is suitable for passenger convenience and safety), then is fed to a signal that the voltage pulse width is reduced by one percent. This operation will result in the cyclical performance of the tram stop directly in the desired route.

Stop at the passenger stop (raising the level of automation control algorithm) will be provided a specific tram waiting time, which is required for passenger service. In the real world this waiting time can controlled by the driver, and this is in passenger security interests, because the main objective of the algorithm, however, is the follow-up passenger stops on time schedule.

Next step will be- move all the objects and their constituents, who will be involved in this control system.

Formulas that will be used in the calculation algorithm:

Tram carried path (m):

$$S_T = X_T \tag{1}$$

Tram braking distance (m):

$$S_{br} = \frac{V_T^{\ 2}}{2 \cdot a} \tag{2}$$

The distance from the tram to the passenger stops (m):

$$S_{TP} = X_P - X_T \tag{3}$$

Breaking time (s):

$$t_{br} = \frac{V_T}{a} \tag{4}$$

Time to stop the motion at current speed V_T, s:

$$t_{kust} = \frac{S_{TP}}{V_T} \tag{5}$$

Braking distance and the distance to the stop the comparison:

$$S_{br} \le S_{TP} \tag{6}$$

The remaining travel time to the next passenger stop t_{atl}, s:

$$t_{atl} = t_P - t_{tekošais} \tag{7}$$

Required speed V_{nepiec}, m/s:

$$V_{nepiec} = \frac{S_{TP}}{atl} \quad (8)$$

Formulas for obtaining the required parameters of the tram (formulas for calculating the parameters of trams require full algorithm development. In reality, those parameters can be obtained from the measuring devices, with the result will not be necessary to calculate these values control program).

Engine speed ω, min^{-1}:

$$\omega = (U_{imp} - 40) \cdot 70 \quad (9)$$

Simplify the calculation it is assumed that at 40% pulse width of the engine speed is already equal to zero, but at 100% of the voltage pulse width speed is 2000 min^{-1}.

V_T tram speed, km / h:

$$V_T = \frac{\omega \cdot 2\pi \cdot R}{R_p} \quad (10)$$

where R – the tram wheel radius, m;

Rp – reducer gear ratio $R_p = 7{,}36$

Tram speed V_T, m/s:

$$V_{Tms} = V_{Tkm/h} \cdot k \quad (11)$$

where k – the calculated ratio $k = 0{,}27$

3. Algorithm for the control system

Before the algorithm to list the building has not provided the variables that will be used in algorithm.

Table 1. Variables that will be used in algorithm

Address	Explanation
M0.1	Variable responsible for starting the timer
T38	T38 Timer which counting current time
SM0.1	A condition which is satisfied every time you run the program
VW1998	Variable which denotes the total number of passengers stops
VW998	Variable which denotes the number of the next passenger stop
VD2000	Variable which denotes the coordinates of the passenger stop
VD2004	Variable which denotes the time at which the stop is due to arrive
M0.3	Variable responsible for starting the timer T34
T34	Timer is responsible for the acceleration of the acceleration value
M5.3	Variable responsible for starting the timer
T201	T201 Timer is responsible for the waiting time at passenger stop
M1.3	Variable which denotes the braking process

After finding all the necessary ingredients of the preparation can begin to draw up the algorithm, after which will be built above the management program, which will be implemented in a programmable controller.

Operational algorithm:

Main tram control program

Step 0: Initialization:

$U_{imp} = 41\%$

$i = 1$

Step 1. Read data from the sensors V_T and S_{nobr}

Step 2. If M0.1 $\neq$ 1, start timer T38 to time the time the route will be enough to complete rout

Step 4. Read next stop data from data base X_{Pi}, t_{Pi}

Step 5. If VW1998 < VW998 and if $S_{nobr} \geq X_P$, then i=i+1 VW1998=VW1998 +1

Step 6. Call **Tram control subroutine**

Step 7. Return to Step 1.

Tram control subroutine

Step 1. Calculate remaining distance between vehicle and passenger stop on route S_{TP}

Step 2. Calculate remaining time for arriving in passenger stop – t_{atl}

Step 3. Calculate necessary vehicle speed – V_{nepiec}

Step 4. Calculate necessary breaking time t_{br}

Step 5. Calculate necessary breaking distance between vehicle and passenger stop according current speed - S_{br}

Step 6. If M0.3 $\neq$ 1, then start timer T34 to *3s*

Step 7. If timer is overflow, then set address M0.3=1

Step 8. If M5.3 $\neq$ 1 and if $S_{br} > S_{TP}$, then $U_{imp} = U_{imp} - 1$

Step 9. If M5.3 $\neq$ 1and if $V_T = 0$, then M5.3 = 1 Set

Step 10. If M5.3 = 1, then start timer T201

Step 11. If T201 = 1 and if M5.3 = 1, then $U_{imp} = U_{imp} + 1$

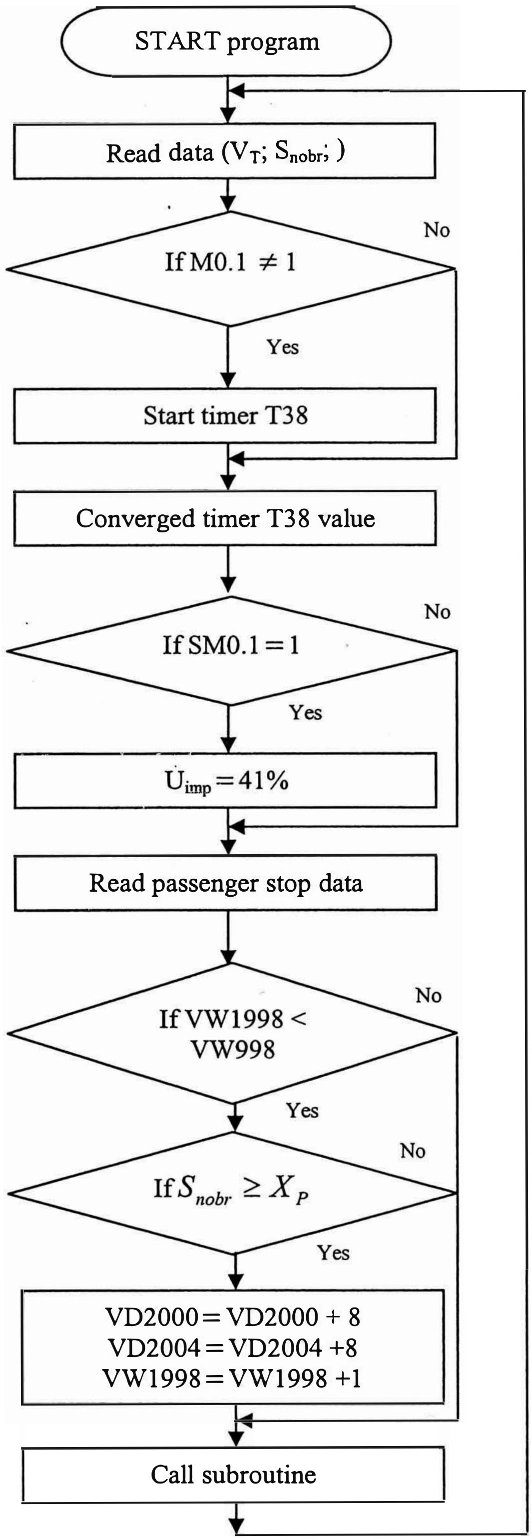
Step 12. If $V_T \geq 0$, then M5.3 = 1 Reset

Step 13. If address and M1.3 $\neq$ 1 and vehicle moving speed is smaller then necessary speed for vehicle and smaller then 20, then increase $U_{imp} = U_{imp} + 1$ (speed increasing)

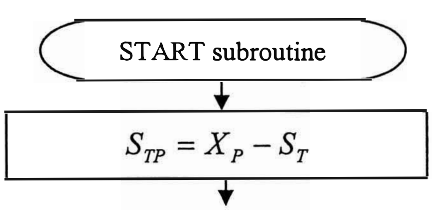
Step 14. END of subroutine

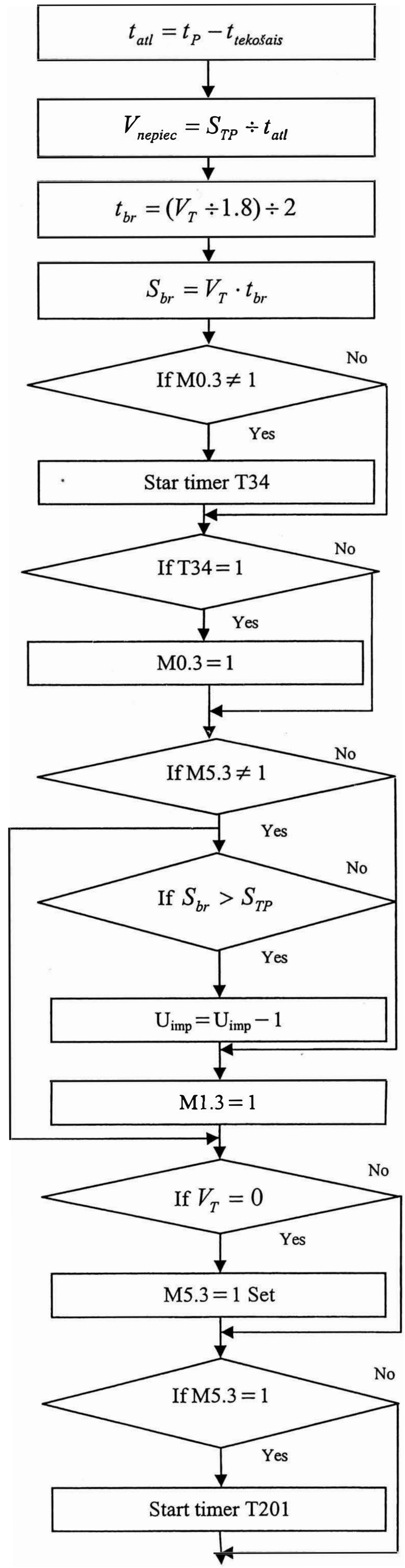
Made algorithm block scheme:

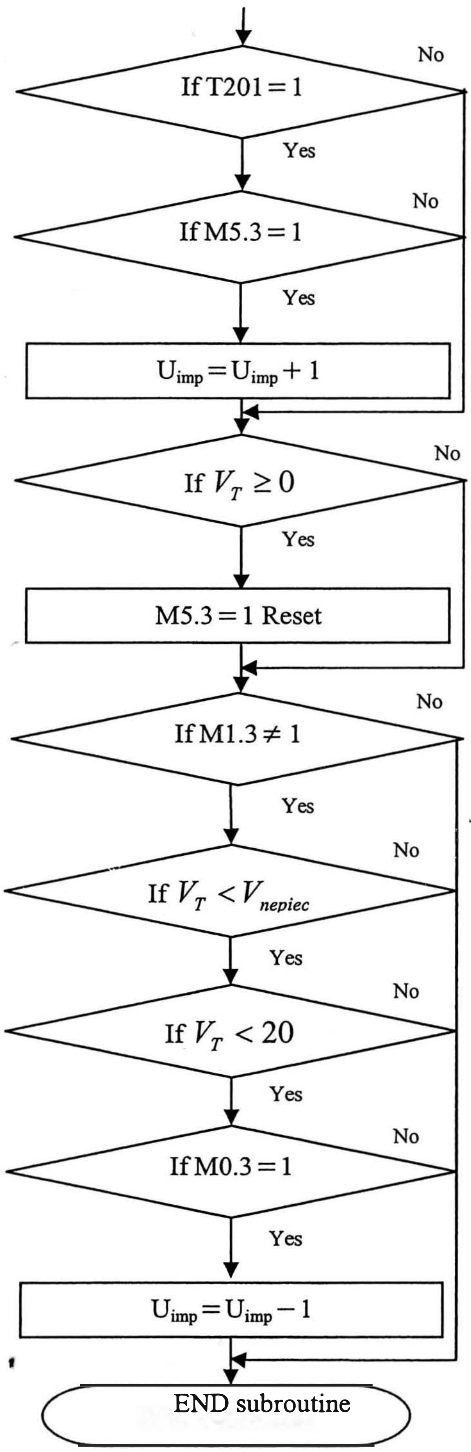
1) Main program block scheme:



2) Subroutine blocks scheme:







4. Algorithm testing

At the time of computer testing during "Step-7" was written an environment program of the described algorithm. Part of the program with the legend is written below.

```
TITLE=SUBROUTINE          Subroutine comments
COMMENTS
Network 1                 If SM0.0 is 1
LD      SM0.0             (always)
MOVR    LD0, VD116        LD0 values for
AENO                      addresses address
-R      VD38, VD116       VD116 and subtract
AENO                      from it the real
AR<     VD38, LD0         value of the address
MOVR    LD4, VD150        VD38 and if the
AENO                      addresses VD38 value
-R      VD72, VD150       is less than
AENO                      addresses LD0 values
MOVR    VD116, VD144      for addresses LD4
AENO                      value and address
/R      VD150, VD144      VD150
AENO                      subtracted from the
=       M1.0              address value VD72
                          and for office
                          addresses VD144 VD116
                          value and
                          it addresses the real
                          value of the address
                          VD150 and
                          M1.0 address for a
                          value of 1
Network 2                 If SM0.0 is 1
LD      SM0.0             (always) VD26 grant
MOVR    VD26, VD136       addresses the real
AENO                      value of the address
/R      1.8, VD136        and VD136
AENO                      distribute it to the
/R      2.0, VD136        value 1.8 and
AENO                      distribute it to 2.0
MOVR    VD26, VD132       and VD26 grant
*R      VD136, VD132      addresses the real
                          value of address
                          VD132 and multiply it
                          with the addresses of
                          the address value
                          VD136
Network 3                 If the address M0.3
LDN     M0.3              value is 1, T34 timer
TON     T34, 30           to run the 0.
Network 4                 If the timer value of
LD      T34               T34 is 1, then assign
=       M0.3              this value to address
                          M0.3
Network 5                 If SM0.0 is 1
LD      SM0.0             (always)
AN      M5.3              M5.3 addresses and
AR>     VD116, 0.0        addresses the value
LPS                       is not equal to 1 and
AR>     VD132, VD116      VD116 address value
-I      +1, VW200         is greater than zero,
AENO                      then
=       M1.3              if the address VD132
LPP                       value is greater than
AR=     VD10, 0.0         the address VD116
S       M5.3, 1           values, reduce
                          addresses VW200 value
                          of 1, and
                          M1.3 or address value
                          is equal to 1
                          then if the addresses
                          VD10 value is equal
                          to 0.0, give the
                          address of a value of
                          1 M5.3
```

As can see, program during the first sub-network using the read from the table, next stop the data is calculated on the distance to the stop, remaining travel time and the necessary speed.

Displayed in the second sub-network is calculated stopping distance at the current rate, while the third and fourth sub-network operations are carried out in connection with the operation of cyclic timer setting the value of which will be use the speed of reception function.

The fifth network while providing the tram brake when it has come close to the station.

The known coordinates of the stops will be given the same program before starting the movement along the route, which will be dispensed without any interchange between the tram and stops. Stop coordinates, or time of arrival changes the case, there is always a possibility to effectively implement the necessary adjustments to the data table.

5. Testing

The real algorithm for checking equipment was involved in bench (fig. 2.) Situated in the industrial programmable controller for Simatic 224xp (1), "ethernet" block (2), the power supply unit (3), fuse (4) router (5) through which only made wireless connections to computers, two traffic lights (6), DC motor (7).

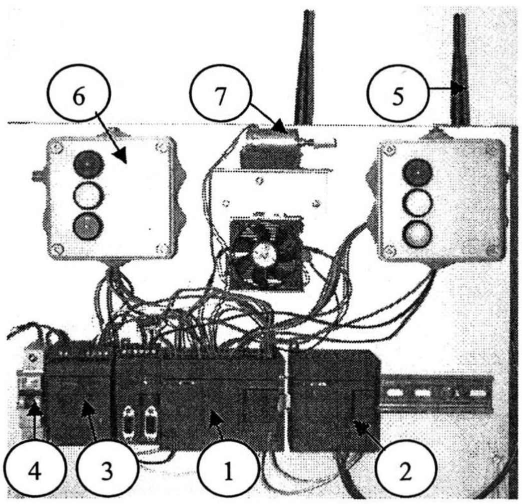


Fig. 2. Stand

When running "Step-7" program was connected to the controller and loaded up the program. “Simulink” medium was prepared by the open model, which graphically depicted the different parameters, both at the tram movements.

This model was connected to the controller with the program “PC Access” assistance, which provided continuous exchange of data between the controller and the model.

Traffic lights signal lights were used in the controller 24 V output.

Engine management was used in the controller output, which generates a voltage with variable pulse width. It was practically the only size that was controlled and enrolled in reality, in other calculations required values were obtained by leaving them directly from this parameter.

6. Conclusions

So can concluded that the established electric tram control algorithm provides accurate tracking of stops on the schedule time by running the tram directly with the speed as necessary to the timely arrival of the stop (accuracy up to seconds).

After a successful start-up program also may be concluded that the control algorithm flowchart is created without errors, because the control program was created by the direct basis of the flowchart.

Experimenting with the stand there’s results, showing algorithm and write the program correctly.

The topic for the development is to refine the algorithm, involving a real device such as sensors that mimic some obstacles arising.

Traffic lights can play a role of passenger stop, which would also stop to them by dividing the total number of routes into small steps, which increase overall program, the precision by reducing the routing number of points, brake which is not provided in the program. Increasing public transport routing speed, in places where it is rational, to be able to manage the traffic lights, regulating signal light so by reducing public transport need to dwell on denying the light signal.

References

1. Ribickis L., Valeinis J. Elektriskā piedziņa mehatronikas sistēmās. Rīga, 2008. – 287 lpp.
2. Ribickis L., Ļevčenkovs A., Gorobecs M. Sistēmu teorijas pamati industriālās elektronikas modelēšanā. Rīga, 2008. – 95 lpp.
3. STEP7.V53 FirstSteps (leaflet) 109 lpp.
4. Levčenkovs A., Gorobecs M., Ribickis L. Programmēšanas tehnoloģijas industriālā elektronikā. Rīga, 2009. – 72 lpp.
5. Ribickis L., Raņķis I., Ļevčenkovs A., Gorobecs M. Programmēšanas valodas industriālā elektronikā. Rīga, 2007. – 69 lpp.